2007

Development and Control of a Solar Array Switching Module

Joseph E. Rymut
Cleveland State University

DEVELOPMENT AND CONTROL OF A SOLAR ARRAY SWITCHING MODULE

JOSEPH E. RYMUT

Bachelor of Science in Electrical Engineering

Cleveland State University

May, 2006

Submitted in partial fulfillment of requirements for the degree

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

at the

CLEVELAND STATE UNIVERSITY

December, 2007

This thesis has been approved
for the Department of Electrical and Computer Engineering
and the College of Graduate Studies by

_____

Dissertation Committee Chairperson, Dr. Zhiqiang Gao

_____

Department/Date

_____

Dr. Lili Dong

_____

Department/Date

_____

Dr. Ana Stankovic

_____

Department/Date

_____

Dr. George Kramerich

_____

Department/Date

DEDICATION

I would like to thank all of my family and friends for there support in the completion of my masters degree.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Zhiqiang Gao, for all of the help that he has given me in the completion of this thesis and all of the opportunities that he has given me in the completion of my masters degree at Cleveland State.

I would like to thank committee members Dr. Lili Dong, Dr. Anna Stankovic, and Dr. George Kramerich for taking the time to read this thesis and being on the committee.

I would like to thank Rob Button at the NASA Glenn Research Center for supporting this research. Also at NASA I would like to thank Jim Fleet for completing the PCB layout and, Dave Houser for populating the circuit boards. Also I need to thank Marcelo Gonzalez for all of his help in this project because without him this project would not have been completed. Last but not least I need to mention Albana Iotova for her work helping complete the system.

In the lab at Cleveland State I need to mention my appreciation for getting to work with Dave Avanesian and Anthony Roberts.

I would also like to thank Adrienne Fox and Jan Basch of the electrical engineering department at Cleveland State for all of the help in doing the behind the scenes work to keep the department running.

DEVELOPMENT AND CONTROL OF A SOLAR ARRAY SWITCHING MODULE

JOSEPH E. RYMUT

ABSTRACT

This research focuses on the development and control of a solar array switching module (SASM). The objective of studying this problem was to develop the SASM hardware and design a controller for the SASM which would effectively deal with the wide ranging dynamics of the system and limit oscillations in the steady state. Initially an intuitive controller was designed to control the SASM. Following this an analysis of the SASM was preformed to create a model which described the SASMs operation. Using the system model, an analysis of a PI and a PII controller was completed which found that both controllers had an undesirable oscillation in the steady state due to the incremental nature of the SASM. To solve this oscillation problem a novel implementation of an integrator is conceived and implemented. Both simulation and hardware test results show that this novel integrator implementation is capable of controlling the SASM without excessive switching.

TABLE OF CONTENTS

Page

NOMENCLATURE

SASM:	Solar array switching module
SCBU:	Series connected boost regulator
PI:	Proportional integral
PMAD:	Power management and distribution
PDU:	Power distribution unit
LEO:	Low earth orbit
PII:	Proportional Integral Integral
SSU:	Sequential switching unit
MOSFET:	Metal oxide semiconductor field effect transistor
GUI:	Graphical user interface
CAN:	Controller area network
OCV:	Open circuit voltage
SCC:	Short circuit current
SAES:	Solar array electrical simulator
PCB:	Printed circuit board

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER I

INTRODUCTION

1.1 Background

Power management and distribution (PMAD) systems are at the heart of any spacecraft, aircraft, or system which relies on electrical power. The PMAD system is a critical component because it is responsible for generating, distributing, and supplying power to all of the systems which require electrical power. The basic topology of spacecraft PMAD systems begins with the power generation sources such as solar arrays, alternators, or nuclear power sources. These power generation sources are then connected to a power distribution unit (PDU) which can be thought of as the distribution center for all of the electrical power. The way in which a PDU operates is that it has electrical sources, loads, and energy storage devices each connected to it so that the PDU can turn on or off the sources, loads, and energy storage components as the system demands change thus controlling the flow of power in the system.

The basic PMAD topology which will be examined as the basis of this research is a topology which consists of a PDU, for power distribution, solar arrays for a source, a lithium ion battery for energy storage, and an unknown dynamic load which is used to simulate the loads in the system. In the overall scheme of this system one other major component which needs to be include is some form of regulation to control the amount of power being supplied from the solar arrays to the load and the battery. This is a critical component to this system because some form of regulation is required to control the charging of the lithium ion battery so that the battery charges properly. The regulator which will be the focus of this research is a solar array switching module (SASM).

The SASM is a regulator which controls the amount of power being supplied from the solar array to the lithium ion battery and the loads. The main objective of the SASM is to control the charging of the lithium ion battery to meet a given charging profile for the battery. It is critical to meet this charging profile because if it is not meet the battery could potentially be severely damaged therefore shortening its useful life or in the worst case completely destroying the battery. The SASM creates the interface between the solar array and the PDU, such that the multiple solar array strings which make up the solar array are each connected to the SASM. The SASM then regulates the amount of power which is being supplied from the solar array strings to the power systems bus in the PDU such that the loads are satisfied and the lithium ion battery is charged properly.

In the area of space power systems there are several different topologies which can be used to regulate the power supplied from solar arrays, or in this case could have been used for the SASM design. The main topology which was discovered in literature is

called a sequential shunt unit (SSU). The SSU topology is a regulator topology where each of the array stings in the solar array is individually attached to a shunt circuit in the regulator. The solar array strings are then shunted if they are not needed, or if the array strings are needed the shunt is removed from the array strings so that it can supply power to the systems bus. Effectively this method works by turning on or off array strings depending on the demands of the system.

In the SSU topology, and any other topology, each solar array string is capable of supplying a certain amount of power to the system where the amount of power supplied from the array strings depends on several different conditions such as operating point, temperature, and lighting conditions. In order to control which array strings are supplying power to the systems bus in a SSU, given all of the different variables which affect the solar array strings, it was discovered that in many SSU applications, a rather simple incremental control circuit is used to control the number of solar array strings supplying power to the bus [1-3]. In the incremental control circuits, many times, a simple comparison of the set point voltage and the actual voltage is made. Then if the voltage is to low, a shunt is removed from a single solar array string to provide extra power to the bus. On the other hand if the voltage is too high an array string is removed from the bus by shunting it out. In order to achieve an effective controller using this method it was found that the control rate has to be fairly fast for this type of regulator because in an application such as the International Space station, the control rate is in the kilohertz range [4].

A second solar array regulator technology which was discovered was a peak power tracking topology. To perform peak power tracking, two different topologies were

discovered the first one being a SCBU topology, where the output of the solar array strings are connected in an SCBU configuration as can be seen in Figure 1 [5]. Once in this configuration the overall output of the SCBU regulator can be adjusted by varying the output voltage of the DC-DC converter and this is generally accomplished by sending a control voltage to the trim pin of the DC-DC converter which can be seen in Figure 1. By controlling the DC-DC converters output voltage several different types of control can be achieved such as a constant voltage mode, a constant current mode, or even peak power tracking. Some of the advantages of this regulator topology include the fact that this design is very efficient in the amount of power it converts and the design can be very modular thus showing that it could be used as a building block for many different applications [5]. Along with discovering the SCBU topology to perform peak power tracking, a second topology which was discovered was a topology in which the output of a solar array string is fed directly into the input of a DC-DC converter where then output of the DC-DC converter is regulated to control the bus voltage at the desired level. The different charging modes which could be accomplished using this configuration include peak power tracking, constant current control, or constant voltage control [6].

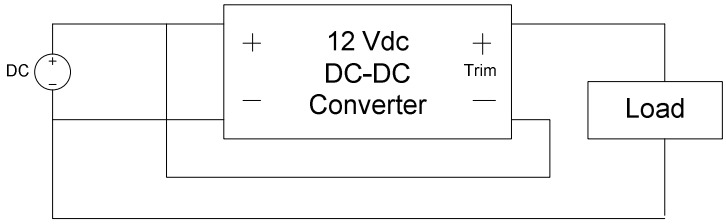


Figure 1: SCBU Configuration

The final topology which was discovered, and the topology which will be studied in the SASM is a series regulator topology [7]. The series regulator topology is a

topology where each solar array string is attached to the power systems bus through a switch such as a MOSFET. The switch is then used to control the power flow from the solar array strings to the power systems bus by open circuiting the solar array string when it is not needed and then connecting the array sting to the bus when more power is needed. In one application where this topology was found the controller was discovered to be a simple voltage controller where solar array strings are turned off as the battery voltage and subsequent state of charge rises [8]. However in the specific application that will be studied for the SASM, there are several different limitations which will prevent using such a simple control topology such as that found in the series regulator of [8] and the different SSU topologies.

1.2 Problem Formulation

The SASM consists of a series regulator topology to control the charging of the lithium ion battery while satisfying the system loads by controlling the amount of power flowing from the solar array strings. In order to accomplish the goal of regulating the charging of the lithium ion battery, the SASM will need to turn on the appropriate number of solar array strings in order to maintain the charging profile for the lithium ion battery. It is critical that the charging profile for the lithium ion battery is meet because if the charging profile is not meet damage to the battery could occur.

The first step in developing this problem and solving it was to design the SASM hardware and integrate it with the other components of the system. This was a critical

first step to be taken because after having the SASM hardware designed and integrated with the rest of the system, this allows for a platform to test different SASM controller designs in hardware. This hardware in turn leads to one of the problems which will be the focus of this research and this is the problem of controlling the SASM. The control of the SASM is an intriguing control problem because the SASM controller has to be able to react to the unknown dynamic load attached to the system while maintaining the proper charging of the lithium ion battery. In designing the controller for this system there are many design considerations and challenges which need to be taken into account such as the changing profile for the lithium ion battery, the unknown dynamics of the solar array strings, and a limited data rate from the battery sensors. These design challenges show that the controller for the SASM will need to be able to operate effectively over a wide range of operating conditions while at the same time maintaining battery charging with a limited feedback data rate.

Another major design criterion for this system is the criteria that the SASM does not continuously turn one solar array string on and off, or this could be thought of as the system oscillating undesirably around the set point. It is desirable to avoid this situation because this constant switching could lead to premature failure of the switches and it also introduces an oscillation into the power system from turning one switch on and off continuously. The situation where one switch is continuously turned on and off appears likely to occur in this system because the system will never be able to exactly achieve the desired set point. This is because the current being supplied from the solar array strings can only be controlled by turning a single channel on or off. Therefore the resolution of the SASM output is limited to output of one channel of the SASM. This could lead to

oscillation from the control system because the controller will never be able to exactly achieve the set point for the SASM but it will continuously try and achieve the set point by overshooting and undershooting the set point which is an undesirable oscillation about the set point or in literature it is also referred to as a limit cycle. This is a major problem because if a simple controller such as a proportional integral (PI) controller were applied to the system it appears as though from intuition there would be some form of oscillation around the set point especially due to the integral action of the controller which would lead to one switch being continuously being turned on and off which is undesirable for this system. As a result, this thesis will explore different ways to control the SASM in order to avoid excessive oscillation in the system while at the same time maintaining good disturbance rejection capabilities.

1.3 Literature Review

The main control problem which will be considered is the problem of avoiding excessive oscillation around a set point due to the fact that only incremental numbers of channels can be turned on in the SASM. In order to try and find a solution to this problem, fist a comprehensive literature search was conducted to see what others have done to try and combat similar problems. The main type of problem which seems most similar to this problem is the problem where there is some type of dead band introduced by the mechanics of a system. This dead band could be introduced into the system by some type of round off error which creates a dead band or relating to mechanical systems

this can be thought of as the problem of precisely trying to control something when there is stiction involved therefore creating a dead band.

In literature it appears as though there are many different proposed solutions to solve the problem of oscillating around the set point or as it is commonly called in literature a limit cycle. One of the most commonly used methods to combat this problem appears to be the introduction of a dead band into the system [9 10]. This method would however not be applicable to this situation because the dead band in the system would need to be static and it would be difficult to determine the range of the dead band. Another method of combating limit cycling which appears to be a common practice is to detune the controller to avoid the oscillation [10]. However this would not be practical in this application because it is desirable to achieve the highest level of performance possible to meet the demands of the system. Some other common techniques which appear in literature appear to be gain scheduling and conditional integration which is where the controllers integrator is stopped or reset within a certain range [10]. It was also discovered that there were techniques which involved knowledge of the plant. These methods include nonlinear compensation techniques [11] and model based techniques [12]. Other techniques which were discovered were techniques which involved a gain margin and phase margin tester [13] and a method which studied pole placement [14]. The different techniques which were discovered in literature appear to show that a wide range of approaches has been taken in trying to solve the problem of limiting oscillations in a system. However these techniques do not appear to be appropriate for this system because all of these methods have some draw back such as needing to know a lot of information about the plant which is not available due to the system dynamics or the

methods proposed are not feasible because they may affect the performance of the system.

1.4 Thesis Outline

The following chapters describe the development of the SASM from hardware design and implementation all the way through controller design implementation and testing. Chapter 2 describes the different components of the system in detail. Chapter 3 then uses the system components to create an intuitive control which is implemented and tested in hardware. Using the test data, and knowledge of the system, next a system model is developed to gain a better understanding of the system and to create a valid simulation to test different controllers. Using the modeling efforts of Chapter 4, Chapter 5 then explores a systematic controller design effort which leads to the introduction and application of a novel integrator implemention which is found to remove steady state oscillations. Lastly Chapter 6 wraps up this research and proposes future work which could be accomplished.

CHAPTER II

SYSTEM OVERVIEW AND HARDWARE DESIGN

The system under consideration is a complete power system for space applications. It is necessary to first understand the entire system in which the SASM will operate in order to fully understand the functioning of the SASM and the requirements which the SASM controller has to meet. The overall system consists of five major components which are the solar arrays, lithium ion battery, PDU, the load, and the SASM which is the focus of this research. First each of these components will be discussed in detail to provide an understanding of there operation and how they interact with the SASM. Having an overview to of the system in which the SASM will operate in, the SASM hardware design is then discussed. With a complete knowledge of all of the hardware involved in this system, the controller requirements for the SASM will be defined.

2.1 System Overview

The main component which binds the entire system together is the PDU as can be seen in the system diagram of Figure 2. From the diagram of Figure 2 it can be seen that the PDU is the component that creates the main bus for the power system. The PDU for this system consists of three relays where each relay is controlled by the graphical user interface (GUI) running on a computer. The GUI can be used to control the operational state of any relay by simply having the user click a button on the computer screen. However during the testing of this system, the PDU relays should remain closed the entire time because there main function in this application will be to provide protection to the equipment. Not only is the GUI used to control the PDU, but it is also responsible for displaying status information on each of the systems, collecting data for analysis, and controlling several functions of the SASM. The different functions which are controlled on the SASM by the GUI are voltage set point, current set point, enabling the SASM, and reporting the status of the SASM.

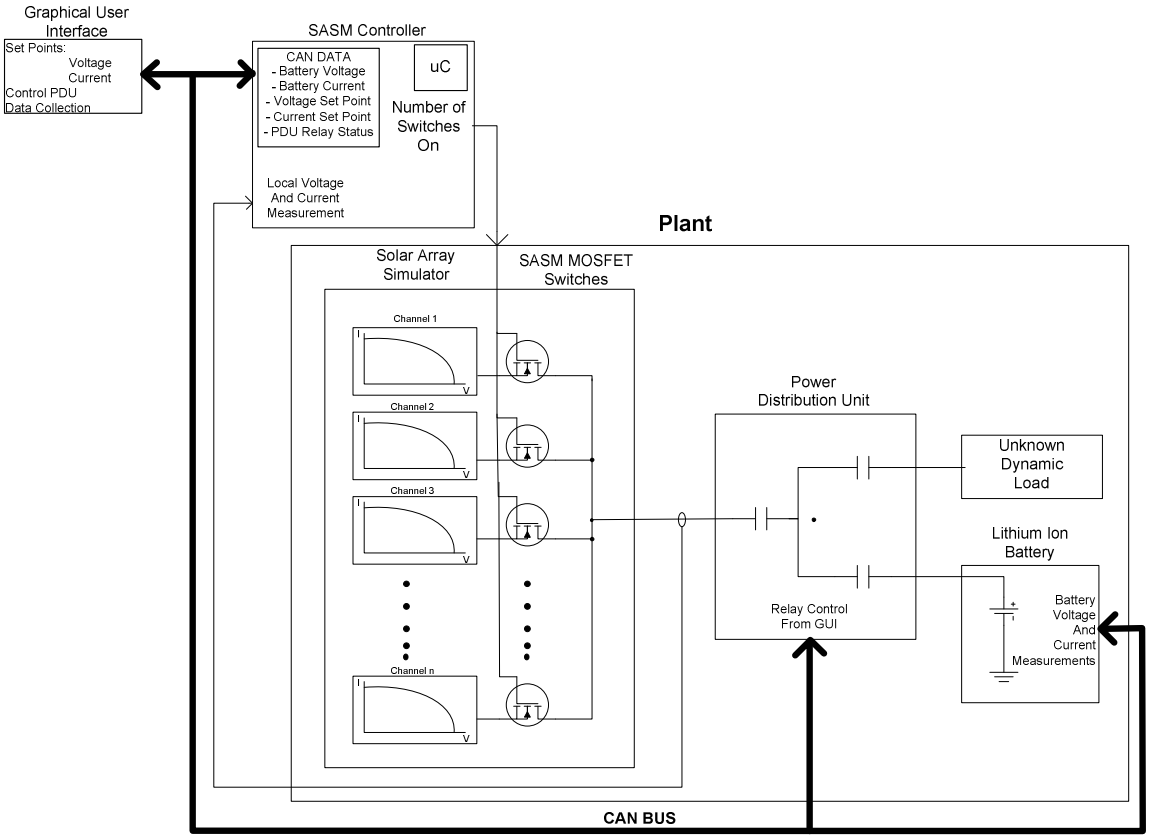


Figure 2: System Overview

In order to facilitate communications between all of these devices a controller area network (CAN) bus is used as can be seen in Figure 2. The CAN communications protocol was used in this application because it is a robust and versatile communications protocol [15]. The CAN communications protocol is a very suitable protocol for this application because all of the devices share a common two wire bus and receive all of the messages sent on the bus. This is useful because each of the nodes, or devices in this case, can be setup to filter out messages such that different nodes on the CAN bus react only to the desired messages for the node [16]. The CAN bus does introduce one limitation in this system and that is the fact that there is a limited bandwidth for the CAN bus. As a result it has to be made certain that the CAN bus can not be overwhelmed with

the number of messages sent on the bus which in the worst case could lead to the crashing of the CAN bus.

2.2 Solar Arrays

The solar arrays are the only power generation source for the entire system. The solar arrays are composed of forty five solar array strings where each array string is capable of supplying the same maximum amount of current which in this case is three amps. The amount of power that each of the solar array strings is capable of generating depends on several different factors. However in order to understand the solar array strings operation, first the general operating principle of the solar array strings needs to be discussed.

Each of the solar array strings operates based on a voltage versus current curve as can be seen in Figure 3 which shows an example of a solar array string operating curve. Every solar array string is similar to Figure 3 because all solar array strings have the same basic shape. Based on this operating curve, an operating point for the solar array string will be established such that the output current will be based on the output voltage of the solar array string which in this case will be the bus voltage.

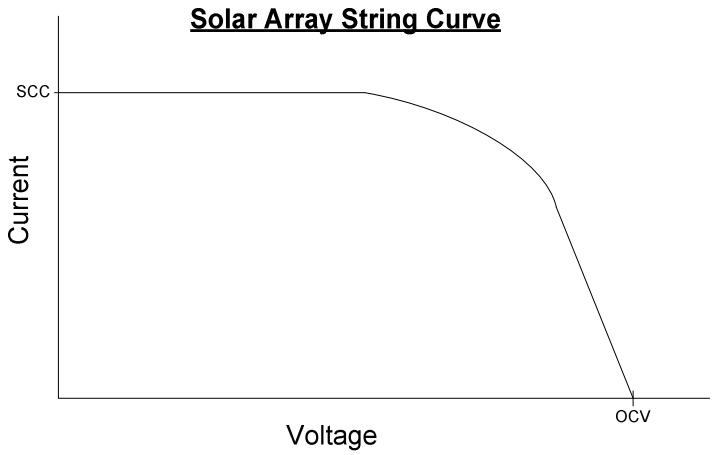


Figure 3: Example Solar Array String Curve

The solar array string curve in Figure 3 is not a static curve; rather it is a dynamic curve. The overall shape of the solar array curve can be defined by two different points. The first point is the open circuit voltage (OCV) which is the voltage where the array string voltage versus current curve crosses the voltage axis. The second important point of the solar array curve is the short circuit current (SCC). The SCC is the amount of current which is generated when the array string is shorted out and this point is defined as the point where the curve crosses the current axis. Depending on the operating conditions of the solar array strings, the OCV and SCC of the solar array changes but the overall shape of the curve remains the same.

The shape of the solar array strings operating curve, which is controlled by the OCV and SCC of the array, is affected by several different conditions including temperature of the solar array strings and the amount of light hitting the solar array. In order to define the dynamics of the solar array strings under real world conditions for this system, it was given that this electrical power system would be used in a low earth orbit (LEO). The LEO profile used in this application is a 90 minute orbit profile where the

system charges the battery for 58 minutes when light is hitting the solar arrays. Then during the remaining 32 minutes of the orbit profile, the battery will be used to supply power to the system thus creating a continuous cycle where the battery is charged and discharged. In order to generate the LEO profile for the solar arrays, without knowing the exact array dynamics for this application, a LEO profile for the International Space Station was scaled to fit the given application because the maximum SCC and OCV were known for the solar array strings in this application. In order to use the International Space Station orbit profiles, the given profiles were scaled down proportionally to fit this application. The resulting OCV and SCC curves which were used can be seen in the plots of Figure 4 and Figure 5 respectively.

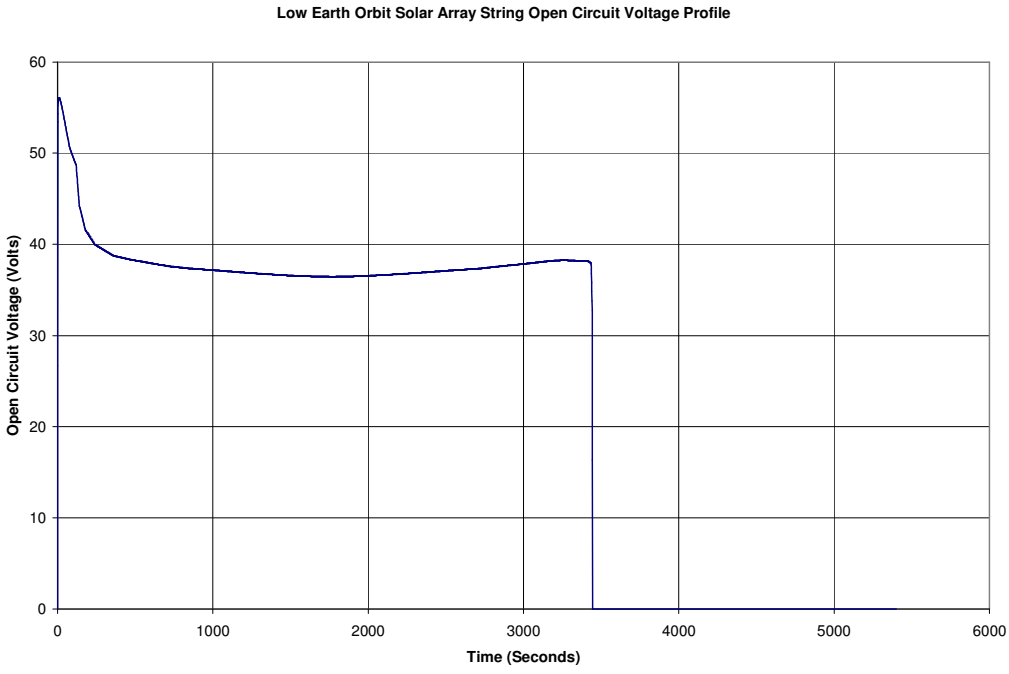


Figure 4: Plot of the OCV for a LEO

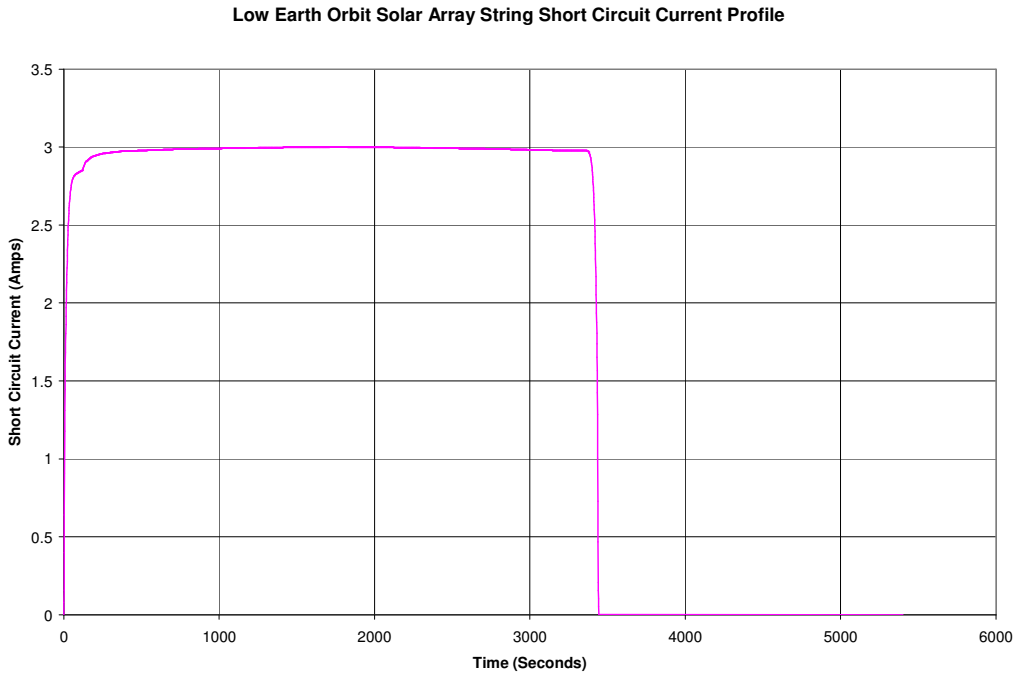


Figure 5: Plot of the SCC for a LEO

In this application, a real solar array is not be used, rather a solar array electrical simulator (SAES) is used. The SAES generates an output which is meant to mimic the output of the real solar arrays. In order to control the SAES so that it mimics a LEO profile, the LEO profile described in Figure 4 and Figure 5 was programmed into the control program of the SAES so that when the SAES is operated it has the same characteristics of a real solar array during a LEO. However one limitation of the SAES which needs to be pointed out is that when testing was completed for the full system, only fifteen solar array strings were completed for the SAES. This is a limitation because this is only a fraction of the 90 solar array strings for which the SASM was originally designed for.

2.3 Battery And Loads

The battery which will be used in this system is a lithium ion battery. This type of battery is a good choice for space applications because it has a low weigh to power ratio, and it does not have memory effects like other battery technologies do [17]. However with the added power that a lithium ion battery can provide from a smaller package this also means that these batteries can become dangerous if improperly charged, over heated, or mistreated. This shows that it is critical to properly charge a lithium ion battery. In order to properly charge a lithium ion battery a basic charging profile needs to be maintained where first the battery is charged in a constant current mode and then once the battery reaches its final voltage, a constant voltage mode takes over which results in a tapering of the charging current to zero when the battery is fully charged. The basic charging profile for the lithium ion battery can be seen in Figure 6.

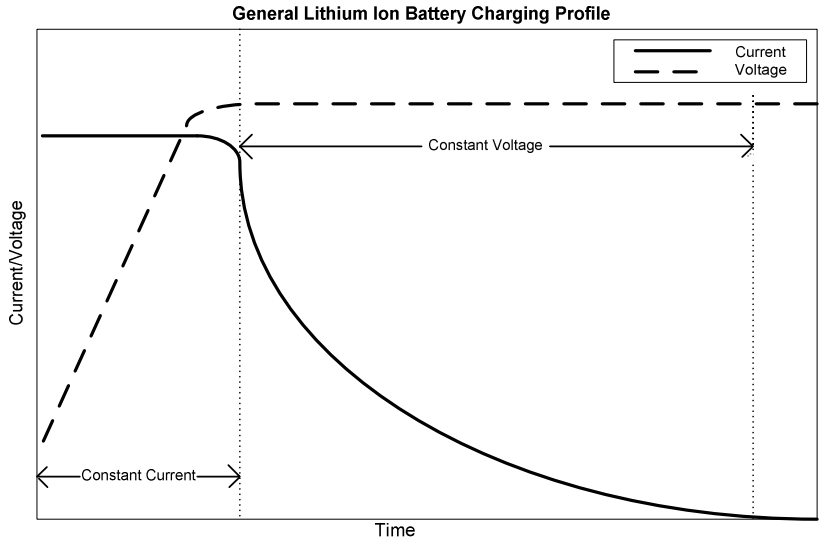


Figure 6: Lithium Ion Battery Charging Profile

The lithium ion battery which was used in this application is a stack of eight cells in series to obtain the desired bus voltage of 33.2 volts for this system. This battery is very unique, because it is actually a “smart” battery which is capable of performing tasks such as monitoring the battery, reporting battery data on the CAN bus, battery protection, and active cell balancing in order to charge all of the cells at the same rate. Since the battery is a “smart” battery it will be used to send batery charging data to the SASM. The data which will be sent to the SASM over the CAN bus is the battery current and the battery voltage. These two pieces of data are critical because they will be used as the feedback in the SASM control loop for charging the battery. However getting the feedback data from the battery over the CAN bus introduces some difficulty into the system design because the battery current measurement and the battery voltage measurement are sent over the CAN bus every tenth of a second. Therefore this will add a limitation into the control system because the control rate can be no faster then the new data coming into the controller thus limiting the control rate to at most 10 hertz.

While controlling the charging of the lithium ion battery, the SASM will also be responsible for supplying power to the load as can be seen in the system diagram of Figure 2. The load in this system is an unknown dynamic load and in this case, the load will be a constant current load. The load in this system is meant to represent the spacecraft system loads such as computers, life support systems, and lighting. In order to simulate these loads, a constant current load bank will be used. The dynamics of loads switching on and off will then be simulated by manually switching on and off the load bank at random times to see if the SASM controller is capable of responding to the unknown load changes. Along with switching the loads on and off, the magnitude of the

loads will also be varied to observe the response of the system. Therefore it can be seen that the load applied to this system is an unknown dynamic load or another way of looking at this from a control systems perspective, the dynamic load can be thought of as an unknown disturbance at the output of the plant.

2.4 SASM Hardware Design

The SASM, which is the main component under consideration in this research, went from a design concept all the way to a finished printed circuit board (PCB) in the development of this project. The main design concept behind the SASM is a series regulator topology which controls the charging of the lithium ion battery and supplies power to the systems bus by turning solar array strings on and off. In the initial design requirements of the SASM there were to be 90 solar array strings to control. Each solar array string would be able to generate a maximum of three amps while the maximum bus voltage would be 33.2 volts which is the fully charged voltage of the lithium ion battery.

One of the major design requirements which influenced the overall design of the SASM was that a modular approach was to be taken in the design so that the systems size could be easily changed and the modular design could add redundancy to the system such that if one of the SASM boards were to fail the other boards could take over and continue supplying power to the system. With these design requirements in place, the basic modular design concept was first completed. In order to make the design modular it was chosen that that each SASM board would be capable of controlling 15 solar array strings.

Using this approach this meant that six different SASM boards would be needed to fulfill the desired goal to control 90 solar array strings. In order to communicate between the different boards and coordinate there control action, the CAN bus was chosen as the communications protocol of choice. The CAN bus was chosen for this system because it was already used as the communications channel between all of the other devices in the system and as a result no matter what communications protocol was to be chosen between the SASM boards, the CAN bus would need to be included in the SASM design because the SASM will be communicating with the battery, the GUI, and the PDU which already send messages over the CAN bus. The overall modular design of the system which was created can be seen in Figure 7 which shows an overview of the SASM system design.

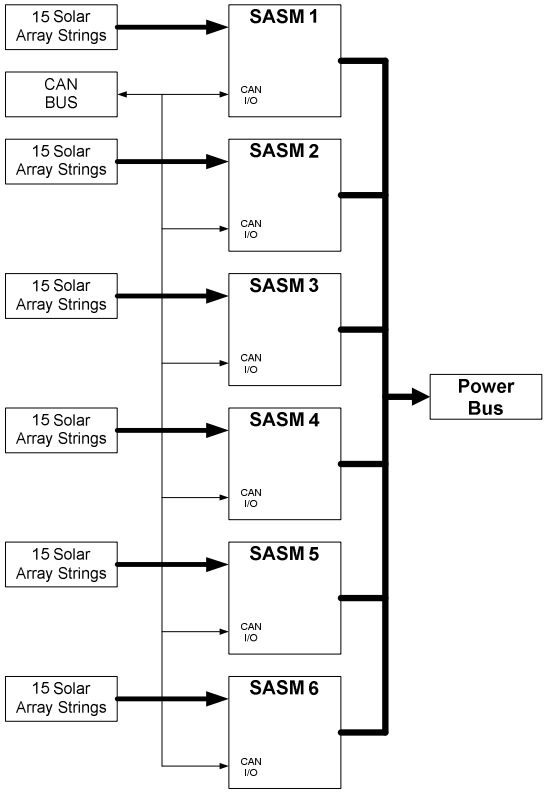


Figure 7: SASM System Structure

Having developed the basic architecture for each SASM board, the next step was to develop the basic circuit design for each solar array string channel. The series regulator topology of the SASM consists of a switch which is placed in series with each solar array string so that the array string can be turned on or off from the power systems bus. The overall topology for one channel of the SASAM can be seen in the drawing of Figure 8. The type of switch which was chosen for this application was a MOSFET switch, as can be seen from Figure 8. In using the MOSFET switch to control power flow there were two design considerations which had to be taken into account. The first is that when using a MOSFET, there is a small on resistance when current is flowing through the MOSFET. As a result the MOSFET for this application was chosen to minimize the on resistance and therefore minimizing the power loss in the MOSFET. In relation to the on resistance the other design consideration which was taken into account was that the MOSFET generates heat when it is turned on. Therefore a thermal analysis was needed for the PCB design of the SASM. In completing the thermal design it was found that the PCB could be designed so that it could be used as a heat sink by making the pads for the MOSFET large enough, since MOSFET used in this application is a surface mount component with a standard SO-8 package [18-20].

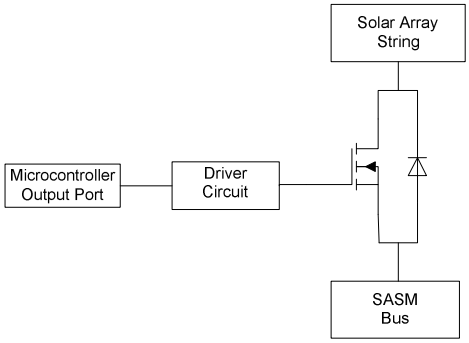


Figure 8: Basic Circuit Design of one Channel of the SASM

With the basic power circuits designed for each channel of the SASM, the next step was to design how the system would be controlled. In order to control each SASM, it was chosen that each SASM board would contain a microcontroller. The microcontroller chosen for this application was a Silicon Labs 8051 family microcontroller [21]. This specific microcontroller is ideally suited for this application because it is easily programmable using the C programming language and it has built in CAN communications capabilities so that it can be directly connected to the existing CAN bus. Another advantage of using this microcontroller is that it is capable of receiving analog inputs. This capability was needed because each SASM board is equipped with a local voltage measurement at its output and it is equipped with a current sensor to measure the total current being supplied by the board. Each of these sensing capabilities was easy to integrate with this microcontroller because only a simple voltage divider circuit is necessary to interface with the microcontroller.

Another advantage of using this microcontroller is that it is capable of controlling the MOSFET switch for each solar array string using the microcontroller's output ports. Each MOSFET switch can be interfaced directly from the microcontroller using a small driver circuit. This driver circuit is necessary because the microcontroller is not capable of supplying the proper gate voltage to trigger the MOSFET. The proper gate voltage for controlling this MOSFET is a voltage which is 12 volts above the source pin of the MOSFET which is also the bus voltage of the SASM output. In order to accomplish this, a DC-DC converter was used in a SCBU configuration so that the gate voltage will float at 12 volts above the bus voltage of the SASM's output. In order to interface between the microcontroller's voltage level, which is 3.3V referenced to ground, and the gate drivers

voltage level which is 12V above the bus voltage, a simple opto-isolator circuit is used as can be seen in the following circuit schematic of Figure 9.

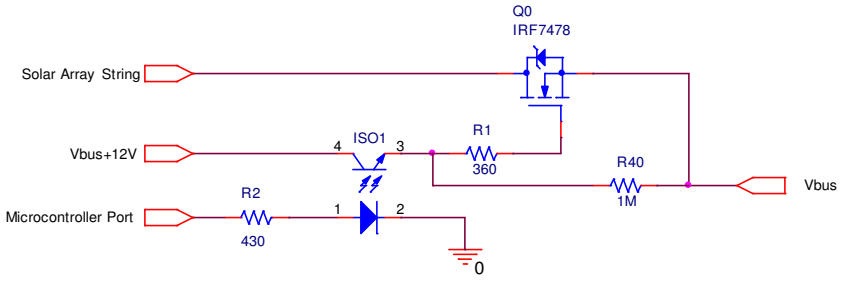


Figure 9: SASM Driver Circuit for one Channel

Having designed all of the basic circuits for the SASM, one other issue had to be taken into account and this issue is what happens if there is no bus voltage. This was a critical design issue because if this condition were to occur, the SASM will not be able to function because it will have no power and because of this the microcontroller will never be able to start or the proper gate drive voltage will never be developed for the MOSFETs. In order to take care of this critical issue a boot strap power supply was added to the system. The boot strap power supply is a power supply which will be present for startup so that the microcontroller and the gate driver can be started up until the bus voltage is developed. The circuit used to do this is a simple ORing diode circuit where one input is the boot strap power supply and the other is the systems bus voltage. As a result, either the bootstrap power supply or the bus voltage will supply power to the system depending on which one has the greater voltage. During normal operating conditions this will be the bus voltage. The completed circuit schematic for one of the SASM printed circuit boards (PCB) can be seen in Appendix A. Having completed the design for the SASM, the final part was to complete the PCB layout for the SASM and

produce the final PCB. The completed SASM circuit boards can be seen in Figure 10 and Figure 11 below.

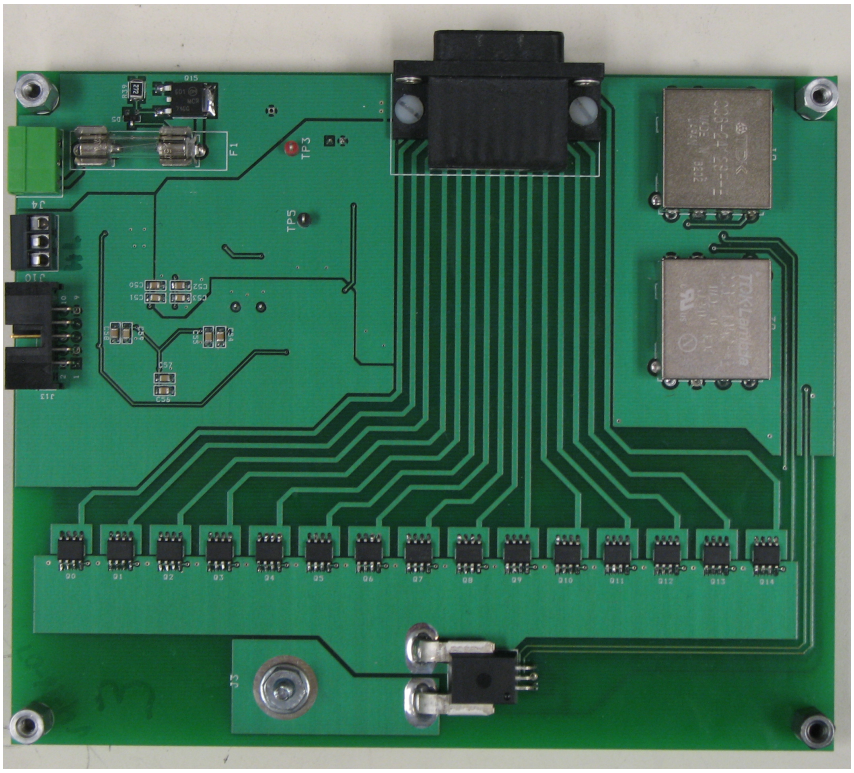


Figure 10: SASM PCB Top Side

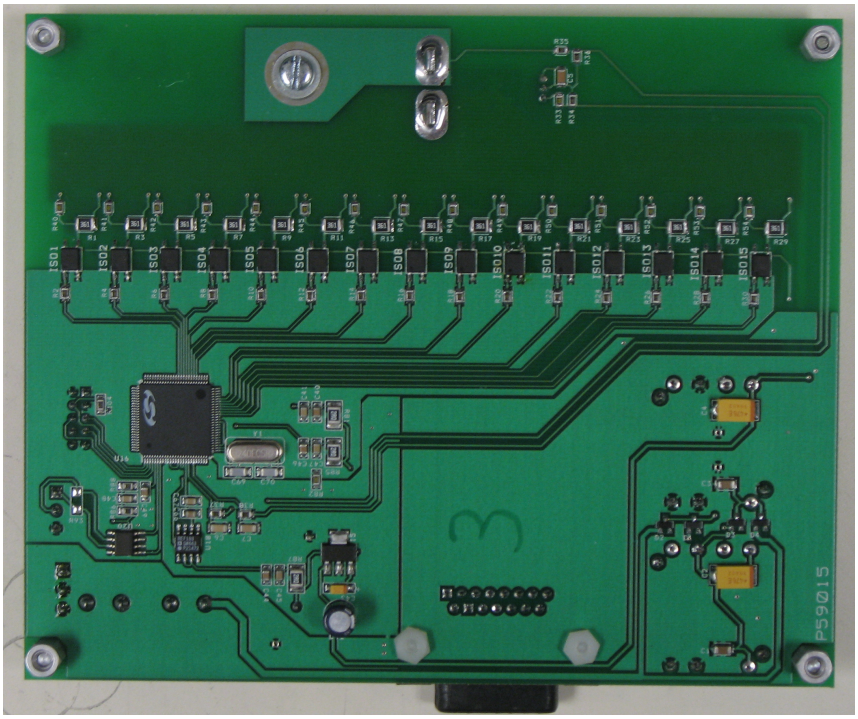


Figure 11: SASM PCB Bottom Side

2.5 SASM Controller Design Goals and Challenges

The overall goal of the SASM controller is to charge the lithium ion battery while supplying power to the other loads of the system. From the desired charging profile for the lithium ion battery which can be seen in Figure 6, this figure shows that the controller for the SASM will require two different control modes one for constant current control mode and another for constant voltage control mode. This shows that one of the design challenges for this controller will be to determine a rule base so that a seamless transition can be made between the two different control modes in the process of charging the battery. Another design challenge which can be seen by the description of the plant

which includes the solar arrays and the batteries is the fact that the plant varies over a wide operating range and its exact operating point is not known or measurable. This adds a challenge into designing the controller because the controller will need to be able to operate over a wide operating range which in the current controllers case means that the current can vary anywhere from zero amps to three amps.

The SASM design in itself adds some difficulty into the design of the controller for this system. The main challenge based on the design of the SASM is that only an integer number of switches in the SASM can be turned on at any time thus introducing a nonlinearity into the system. The fact that only an integer number of switches can be turned on introduces the problem that there could be excessive switching in the system. The excessive switching could come from the fact that the controller will never be able to exactly reach the set point for the system. Therefore it would be desirable to have a controller which does not keep turning a single switch on and off in order to try and reach a set point which is actually never attainable. As a result it would be desirable to have some type of a dead band around the set point to avoid excessive switching. The type of dead band which would be acceptable in this system would be a dead band where the actual output could be within the range of the set point plus and minus the results of turning one channel on or off. For example in the current mode if the set point is 10 amps and the current per channel were 1.5 amps then the acceptable range of operation would be from 8.5 amps to 11.5 amps.

In this system, there will be random unknown load changes introduced into the system such that a load could be turned on or off at any point in time. As a result, it will be the SASM controller's job to react to these load changes, which can be thought of as

disturbances in the output of the system. The desired recovery time for this system to react to disturbances is 0.5 seconds. This does not seem like a very difficult target to reach as far as the controller is concerned however there is one severe limitation on the controllers operating speed. The main limitation on the controller is the fact that the battery current and battery voltage data for the SASM's feedback is sent over the CAN bus at a data rate of 10Hz. This is a severe limitation for the SASM controller because this means that the controller has to operate at a rate of 10Hz or below. In actuality the control rate for the SASM controller should be set below 10Hz in order to avoid timing issues which could arise from delays in the CAN bus due to the fact that certain messages may have priority over the battery data on the CAN bus therefore adding an unknown delay to the data sent from the battery. As a result, the target control rate in this application will be 8Hz which is assuredly low enough to avoid any timing issues involved in receiving new data over the CAN bus.

An interesting analysis which can be preformed at this point is to show the difficulty in trying to react to changes in the system by the SASM in 0.5 seconds with a control rate of 8Hz. The analysis which will be preformed is the calculation of the lowest desired control rate for this system using a generally accepted and widely found rule of thumb that the minimum control rate should be ten times the bandwidth of the system. Using the desired response time for the system and knowledge that the desired response is a first order response, the bandwidth of the system can be calculated to be 1.27Hz using Eq. (3.1).

$$Band\ Width = \frac{2}{\pi * Rise\ Time} \tag{3.1}$$

Using the rule of thumb, that the minimum control rate for the system should be 10 times the bandwidth of the system, this leads to the minimum control rate for the system which is 12.7Hz. As a result it can be seen the desired control rate of 8Hz for the SASM will be below the rule of thumb minimum control rate thus showing an added difficulty in controlling the SASM.

Another challenge in designing the SASM controller is the fact that the SASM is actually distributed over several different boards. As a result some type of distributed control system needs to be developed for the SASM. A limitation placed on the distributed nature of this control system is that the CAN bus has a limited bandwidth, which means that the number of messages sent over the CAN bus between the different SASM boards should be limited in order to not overwhelm the CAN bus.

CHAPTER III

INITIAL CONTROL DESIGN, IMPLEMENTATION, AND TESTING

With the requirements for the SASM controller defined and a basic knowledge of the system, the next step was to try and close the loop in the system and design an initial controller for the SASM. The initial controller, which was designed based on a rough knowledge of the system, was an intuitive controller. After designing the intuitive controller, the next step was to implement the controller in the SASM hardware where the first step in implementing the controller was to design the common microcontroller functionalities which will be used in every controller implemented in the SASM. After implementing the intuitive controller in the SASM, hardware tests were completed to validate the functionality of the intuitive controller in a LEO.

3.1 Intuitive Controller Design

The initial controller which was designed and studied to control the SASM was an intuitive controller. This controller was designed first because at the time of designing this initial controller there was little information known about system such that an accurate system model for the system could not be developed. The main reason for this lack of knowledge came from the fact that the other system components were not fully completed at this time and there was no way to know the interaction among them. Overall the intuitive controller is very simple in the way in which it operates. The intuitive controller can be thought of as a proportional controller where the gains are adjusted based on information obtained from the operating conditions of the system. The intuitive controller for the SASM can be broken down to have two different control modes the first control mode is a constant current control mode and the second control mode is a constant voltage control mode. These two different control modes are needed because initially the battery will start in constant current mode and then transition into constant voltage mode as the voltage of the battery rises to the set point therefore meeting the charging profile for the lithium ion battery which can be seen in Figure 6.

The first control mode which was studied was the constant current control mode. The constant current control mode can be thought of in a very intuitive way if we are able to measure the total current from the SASM and if the total number of SASM channels turned on is known. Knowing these two pieces of information and working on the assumption that all of the solar array strings have approximately the same OCV and SCC it can be assumed that all of the solar array strings have the same current versus voltage

curve. This first assumption is a valid assumption to make because all of the solar array strings are exposed to the same operating conditions thus showing that the characteristic curves for the solar array strings will be the same. With this assumption in place, next it can then be inferred that all of the solar array strings will have approximately the same operating point on the solar arrays current versus voltage curve and that the operating point will remain approximately the same when additional solar array strings are turned on or off. This is a valid assumption to make because the battery will hold the bus voltage at approximately the same point even when additional array strings are turned on or off thus showing that the operating point for the solar array strings will remain approximately the same. Working with these assumptions, the rules for calculating the gains of the controller can be obtained.

The basic control loop for the constant current controller can be seen in Figure 12. From this control loop in Figure 12 it can be seen that the control approach is an incremental control approach where the change in the number of SASM channels is determined based on the error signal generated. In order to determine the change in the number of switches this is where the calculated gain, K, comes into the controller. A simple way of looking at the controller gain K is to think of it in terms of unit conversion. If the error is in amps and the change in the number of switches is an integer, then the units of the conversion factor between the error and the change in the number of switches is the number of switches per amp. This factor is readily available because the total current being supplied by the SASM is measured and the total number of switches on is known by the controller. This in turn leads to the controller gain K because this gain can be calculated by dividing the number of channels on by the total current being supplied

by the SASM. Next it should be noted that there is a rounding block after the summing junction of the incrementer as can be seen in Figure 12. This rounding block is in place because only integer number of switches can be turned on at one time. This rounding is also advantageous because it acts as a limiter in order to help avoid excessive switching because a channel will not turn on or off until the threshold of the rounding block is reached or in physical terms until a single channel can be turned on or off.

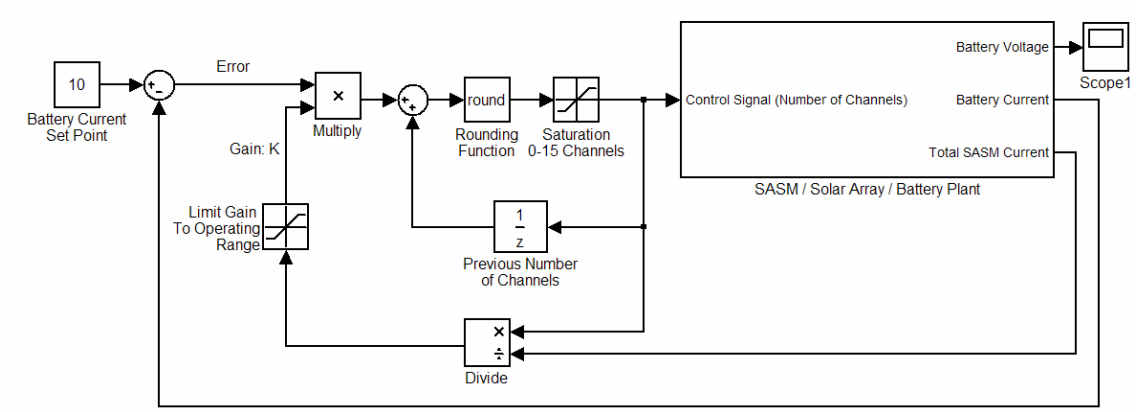


Figure 12: Basic Constant Current Control Loop

The constant voltage portion of the control loop was designed in a very similar manner to the constant current control loop. The constant voltage control loop can be thought of in the same manner as the constant current control loop in Figure 12, where the only difference is the voltage set point is the maximum voltage of the battery and the feedback signal is the battery voltage. The gain in the voltage controller will be called K_v and if it is looked at from a unit conversion stand point the gain K_v needs to be the number of channels per volt. Since this calculation can not be directly computed as for the constant current control mode, a somewhat indirect method is used. To measure the change in voltage per channel, the voltage change is measured each time a channel is

turned on or off. This in turn leads to a calculation of K_v because the change in voltage per channel can be calculated and if the inverse is taken, this will lead to K_v which is the number of channels per volt. This method can be used to calculate K_v because the changes in the voltage mode are much smaller then the changes in current which occur. This is important because it allows for the calculated value of K_v to be accurate over a wide range of operation.

Having designed two individual controllers for the SASM, the next step was to combine the two controllers so that there is a seamless transition between the two different control modes. In order to combine the two different controllers the different conditions of the battery and the plant were studied during a charging cycle in order to come up with a rule base. From studying the plant it can be seen that if the controller with the smaller change in the number of SASM channels on is chosen, the controller will operate properly. This rule base is adequate because if the charging profile is studied, first the current control mode will take over because the voltage error will be very large thus resulting in the voltage controller generating a control signal which will command a large change in the number of channels. On the other hand, the error in the current control mode will be much smaller thus generating a smaller change in the number of channels which will lead to this control signal being the dominant control signal. As the battery charges in the constant current mode, the change in the number of switches, for the voltage control mode decreases because the battery voltage rises as the battery charges thus decreasing the battery voltage error. At some point there will then be an equilibrium in the change in the number of channels on between the two controllers. Then the voltage controller will eventually take over because voltage set point will be

reached thus limiting the error in the voltage controller which in turn limits the change in the number of switches on due to the voltage control mode. The rule base which was described can be seen in the flow chart of Figure 13.

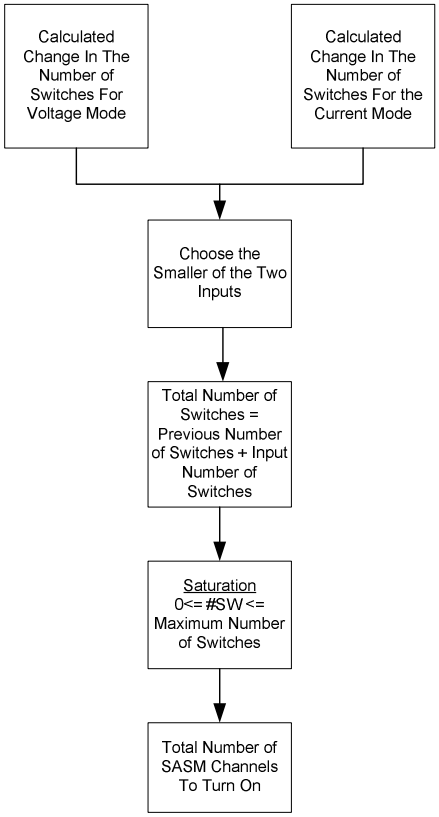


Figure 13: Intuitive Control Mode Decision

3.2 Common Hardware and Software Implementation Issues

Having designed the intuitive controller for the SASM, the next step in the development process was to implement the intuitive controller in the SASM hardware so that the effectiveness of the controller could be tested in hardware. Also it was important

to complete the step of implementing this SASM controller in hardware because the results will in the end lead to an analytical model for the system. However prior to implementing the intuitive controller in hardware, some design issues concerning the microcontroller and the distributed nature of the control system need to be discussed. These design issues are very important because they will lead to common software for the SASM microcontrollers which will be a part of every controller implemented in the SASM.

The first the issue for the SASM controller which needs to be taken into account is the issue of coordinating between the different SASM boards in order to create a distributed control system. In order to keep the controller as simple as possible for the initial implementation of the SASM, it was decided that a simple coordination scheme would be used where one SASM board is the master and the other SASM boards in the system are slaves. This type of distributed control scheme was chosen because it avoids many issues involved in creating and implementing a more complex distributed control system even though it may be desirable in the future to implement a control system which is more complex in order to incorporate redundancy in the system.

The way in which this master-slave configuration works is that the master SASM will be in charge of all of the control activities such that it will send messages to the slave SASM boards over the CAN bus so that they will perform the necessary tasks for the master SASM such as turning on channels and reporting back operational data. The way in which this scheme is set up is that the master SASM microcontroller is set up so that it has a simple timing loop which generates an interrupt at the controller frequency and half the controller frequency. So effectively the controller generates an interrupt at twice the

control rate. What occurs at the first interrupt is that the master SASM sends out a request for data over the CAN bus so that the slave SASM boards will send back updated current and voltage measurements to the master SASM which will be performing the control activities. At the next interrupt, using the data collected from the previous interrupt along with the most current data from the battery, which is also received via the CAN bus, the new number of channels on is calculated by the master SASM. After doing this the slave SASM boards are sent a CAN message which instructs them on how many channels to turn on. After completing this, the same control cycle is repeated over and over. This structure also has some robust features built into it which were not fully implemented for simplicity at this time. The main feature is the fact that any SASM can be the master SASM because they are all programmed the same. However the ability to switch between which SASM is the master has not been fully implemented even though the ground work has been laid in the software and the CAN messages used to communicate between the SASM boards. This feature would add a level of robustness to the system because if one of the SASM boards were to fail this could be recognized by the other SASM boards in the system which would then allow the remaining SASM boards to continue to operate in some limited capacity.

Another piece of common software, shared between the different controllers, is the part of the software that determines which switches on what boards are turned on or off depending on the number of channels demanded. In order to determine which switches are on or off, it was decided that a round robin approach would be taken in making this decision. What this means is that for example if there were six boards and it was desirable to turn on 10 switches, boards one through four would have two channels

on and the remaining boards would only have one channel on. So as it can be seen from the example each board has one channel turned on until the last board is reached and then the process is repeated until the total number of switches required is met. This type of method for turning the channels on was chosen because it spreads out the heating of the boards evenly among the SASMs thus minimizing the heat generated in any one board.

A related house keeping task, which was included in the implementation of the common portion of the SASM controller, was the inclusion of the capability to control the SASM and monitor its operation from the GUI. This part of the microcontroller code is a common piece of software between any controller implemented in the SASM because these functions would need to be present no matter what controller is applied to the system. In order to add this functionality, the CAN bus was setup to include messages which tell the SASM when to send status information to the GUI along with messages to command certain functions of the SASM such as setting the set points or enabling and disabling the SASM. This functionality is very useful from a user's stand point because it allows simple changes, such as changing the set point, to be made in the SASM without reprogramming it and it allows for the SASM to send data to the GUI for collection and later analysis. In order to more clearly understand the software functionality which was described, a flow chart for all of the common microcontroller code can be seen in Figure 14. One block which needs to be pointed out is the controller calculation block. This block is important because it is the block where all of the controllers will be implemented.

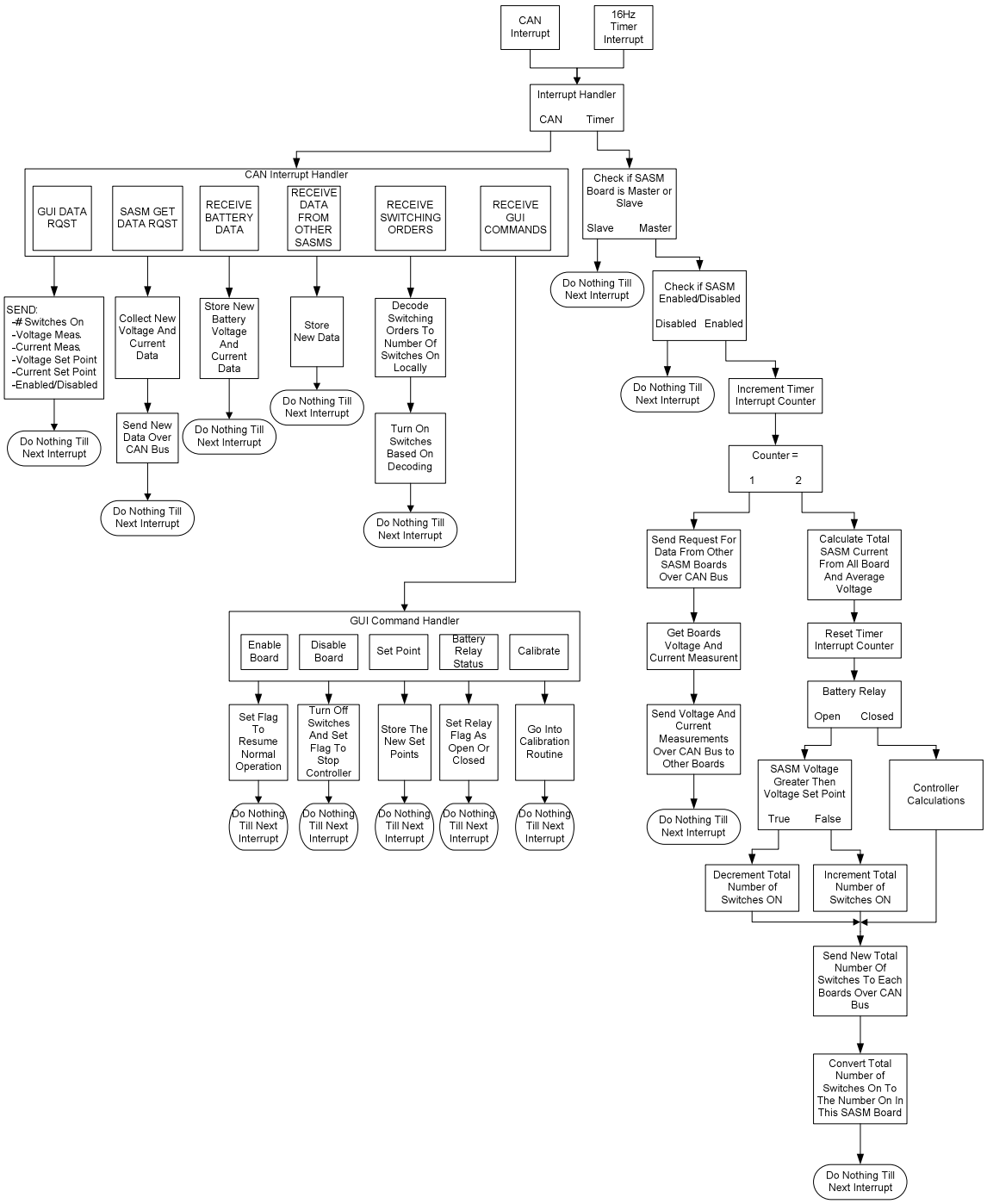


Figure 14: Overall Microcontroller Software Structure

A major hardware limitation which should be mentioned at this point is the fact that currently the SAES has only fifteen channels available for testing. This is a limitation on the system to be tested because the initial SASM design included the

capability to take in ninety channels from the SAES. This in turn places some limitations on the LEO cycle testing of the system because the battery can only be drawn down to ninety percent state of charge as this was calculated to be the lowest the state of charge that could be attained while still allowing the system to fully charge during the LEO with the limited number of channels available in the system and at the same time allow for the transient capabilities of the controller to be tested. This limited number of channels also limited the number of SASM boards in the system to two boards rather then the expected six SASM boards which were planned in the initial design.

3.3 Hardware Implementation and Results

Having defined the basic structure to implement any controller in the SASM hardware, the next step was to implement the control structure for the intuitive controller. The intuitive controller structure which was implemented follows the Simulink block diagram of Figure 12 and can be seen in Figure 15. The way in which this controller operates is that during each iteration of the controller, a new gain for the constant current mode is calculated, and the new gain for the voltage control mode is calculated if the number of switches has changed. Next the controller calculates the error for each control mode and calculates a change in the number of switches for both the voltage control mode and the current control mode as can be seen in the block diagram of Figure 15. Lastly which control mode to go into is determined using the logic previously described

in Figure 13 which chooses the control mode which has the fewer number of switches to change which then leads to the total number of channels to turn on in the SASM.

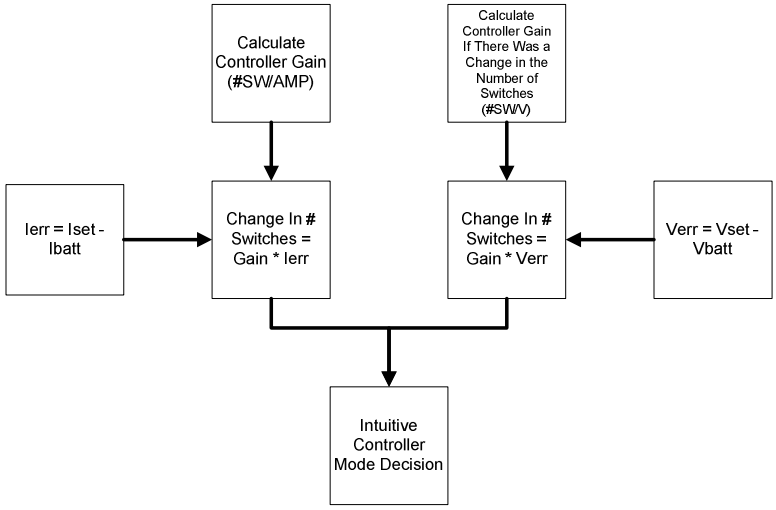


Figure 15: Controller Structure

Once the code for the microcontroller was developed and implemented, the next step was to test the controller in the system for overall functionality. The first test which was developed for this controller was to perform a LEO test using the LEO profile described in Figures 4 and 5. In the first LEO test which was preformed, the battery, which was initially fully charged, had a ten amp constant current load applied to it in order to discharge the battery to 90 percent state of charge. A load of this magnitude was applied to the system because it was calculated that this would result in a 90 percent state of charge if it was applied for 32 minutes. This was the target state of charge because with the limited number of channels available, the SASM would be able to fully charge the battery in the given 58 minutes of illumination. After discharging the battery at the given rate for the total eclipse time, the load was reduced to three amps and the SAES was started so that the simulator was just coming out of eclipse. Then the SASM and the

entire system was run for the entire period of illumination in order to charge the battery using the SASM controller. Next when eclipse was reached, the constant current load was again increased to 10 amps for the entire eclipse period. Following this the battery was again charged and this same process was completed for several orbit cycles to study the effectiveness of the SASM regulator in charging the battery in a continuous LEO.

After successfully completing several LEO's the next step was to observe the disturbance rejection characteristics of the system. To observe these effects, additional constant current loads were toggled on and off several times during the simulation. While doing this the responses of the battery current and battery voltage were observed on the oscilloscope in order to measure the response time of the controller. The load magnitudes and the times to apply the step changes in the load were determined by studying the charging profile of one of the LEO's which was preformed. From doing this, the best times to apply load changes were determined by looking at the number of extra switches that could be turned on in the system. In doing this times were picked where there was room for extra switches to be turned on. These times were chosen because the controller's action could be studied if load changes of an appropriate magnitude were chosen such that the total number of switches is not reached because if the total number of switches was reached the effectiveness of the controller could not be studied.

Having preformed the tests described to observe the effectiveness of the SASM controller, the next step was to examine the results which were produced. The data which was generated to analyze the results was mainly captured using the GUI which has the capabilities to store all of the data generated by the different devices in the system

attached to the CAN bus. Using the data which was obtained, several different plots could be generated as can be seen in Figure 16 which show of one of the charging cycles for the lithium ion battery during the continuous LEO test. Figure 16 only shows one of the cycles which is representative of all of the charging cycles which were completed.

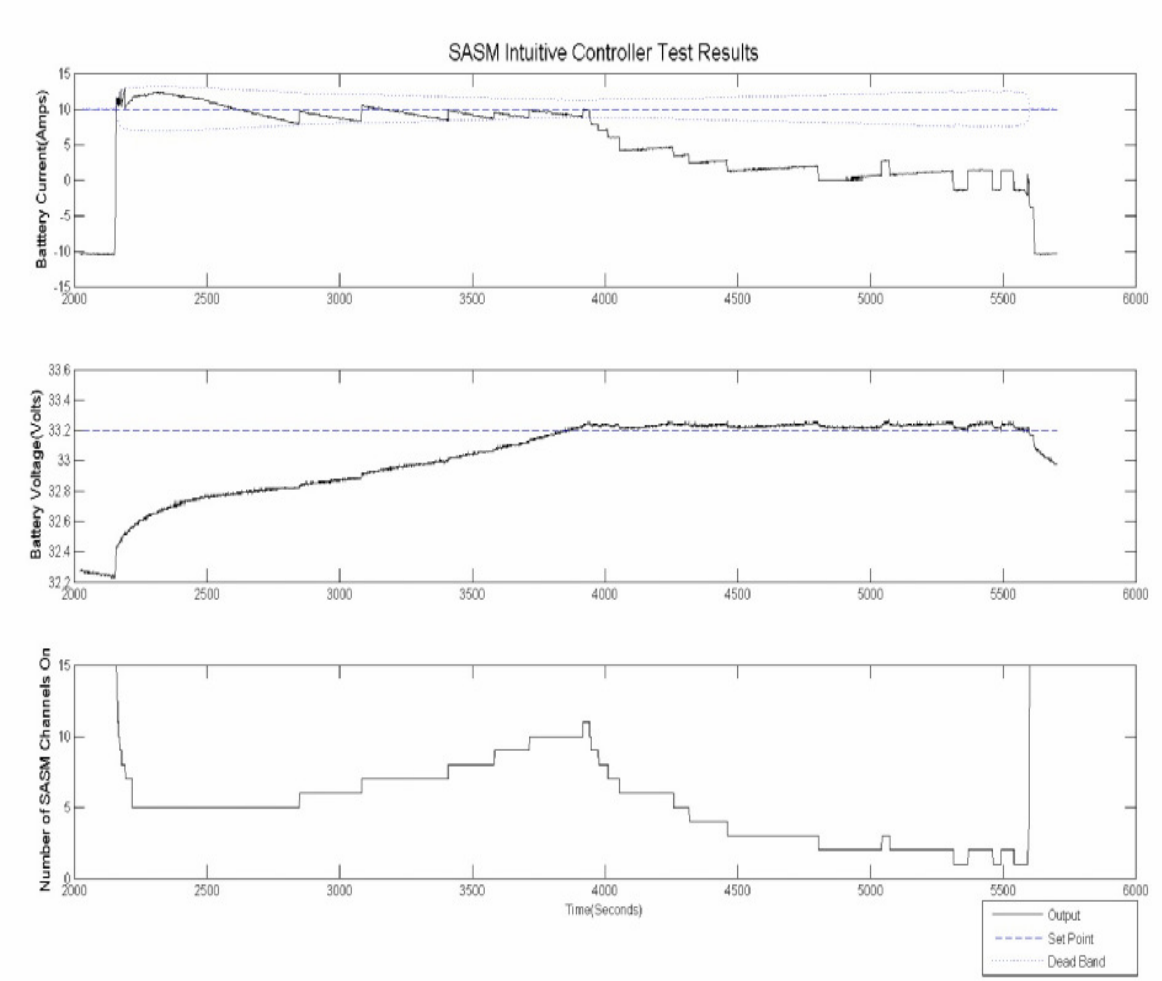


Figure 16: Hardware Test Results

The data of Figure 16 shows that the intuitive controller in the SASM is capable of controlling the charging of the lithium ion battery. This can be seen because if the results of Figure 16 are compared with the desired charging profile for the lithium ion battery in Figure 6 they follow the desired charging profile. The only notable difference

between the results obtained and the desired charging profile is that the test results show an overall saw tooth pattern in the current and voltage measurements. This however is expected because the SASM does not have infinite control rather it is more incremental and can only turn on or off channels which are very coarse in nature. These results also appear acceptable because the problem of excessive switching appears to be avoided in the results, as this was a design concern and one of the control goals set forth in the design process. One small problem which shows up in this controller is the fact that the current and the voltage do not appear to create an average around the set point. As a result if the current or the voltage were averaged, the computed average would not equal the desired set point. This observation is confirmed by the fact that if the constant current portion of the results in Figure 16 are observed, the battery current appears to sit below the set point most of the time thus showing that the average value of the current will never be equal to the set point. However this is sill an acceptable output because the output is sitting within the dead band which was defined earlier in the system requirements. As a result, this shows that even thought the controller operates acceptably there is still some room for improvement in the controller to try and achieve a true average equal to the set point.

After showing that the SASM is capable of properly charging the lithium ion battery by following the proper charging profile, the next step was to see if the disturbance rejection capabilities of the controller were sufficient for this application. The results which were obtained for step changes in the load, which were observed on the oscilloscope, can be seen in Table 1 which shows the response times for the intuitive controller for different load changes. The desired goal for the system was to obtain a

desired response time of 500 milliseconds. From the averaged results in Table 1 for the intuitive controller, it appears as though average response time is within the target response time. However from studying the results it appears as thought there are two outliers in the voltage control mode where there response time is greater then the goal of 500 milliseconds. These outliers are however still within an acceptable range for this system because the results show that these outliers are within one controller cycle of the desired response time. This result therefore shows that there is some room for improvement in the system in the means of disturbance rejection even though the results are very good considering a 500 milliseconds response time is only four controller iterations. Overall, as a result of testing the intuitive controller, it appears as thought there is room for improvement to try and achieve a true average output equal to the set point and to try and improve the disturbance rejection capabilities of the system.

TABLE I: CONTROLLER STEP RESPONSE RESULTS

	Step Time (Min)	Load Type	Load Magnitude (A)	Intuitive Control Response Time (sec)	Incremental PI Control Response Time (msec)
Current Control Mode	4	On	15	180	125
	5	Off	15	110	190
	9	On	15	210	380
	10	Off	15	260	370
	14	On	10	280	380
	15	Off	10	330	440
	Average Response Time =			**228.333**	**314.167**
Voltage Control Mode	38	On	8	280	460
	39	Off	8	470	390
	42	On	15	480	640
	43	Off	15	500	740
	45	On	15	320	620
	46	Off	15	600	620
	52	On	15	480	560
	53	Off	15	640	510
	Average Response Time =			**471.250**	**567.500**

CHAPTER IV

SYSTEM MODELING AND VERIFICATION

To try and improve the controller for the SASM from the intuitive controller, the first step was to gain a better understanding of the system using a more systematic approach. In order to gain a more complete knowledge of the system, the first step was to try and model the system so that, a simple transfer function model could be derived for the plant. After deriving a model for the plant based on test data from the system, next the other components of the system will be modeled based on the knowledge of the system and test data which was obtained. Lastly, this chapter will take the model created and apply the intuitive controller to the model so that a comparison can be made between the system model created and the test results which were obtained.

4.1 System Model

The first portion of the system which was modeled was the plant which is composed of the SASM and the solar arrays. It is important to model this portion of the system first because it would be desirable to obtain a transfer function for the system so that a thorough controller analysis could be completed for future controller designs. In order to observe the response of the system, the first test which was preformed was to find the step response of the SASM so that from this response, a transfer function for the system could be derived.

In order to generate a step response for the SASM, the SASM was programmed to go from having zero channels turned on to having on one channel tuned on. While performing this test, the SCC of the SAES was set to the maximum value of three amps and the OCV was set to 45 volts. The SAES was setup in this manner so that the operating point of the solar array strings would end up in the horizontal constant current portion of the curve. A operating point on the constant current portion of the curve would occur because the battery would be attached to the systems bus during this test so that the operating point of the system would be set to be bellow 33.2 volts which will definitely end up in the constant current portion of the solar arrays operating curve. With these initial settings, the step response of the system was performed and the resulting response of the current flowing through the channel turned on in the SASM was captured on the oscilloscope as can be seen in Figure 17. Next while holding the OCV constant, the step response of the system was also observed for other SCC values below three amps

and it was found that the magnitude of the steady state current changed in direct proportion to the SCC of the SAES.

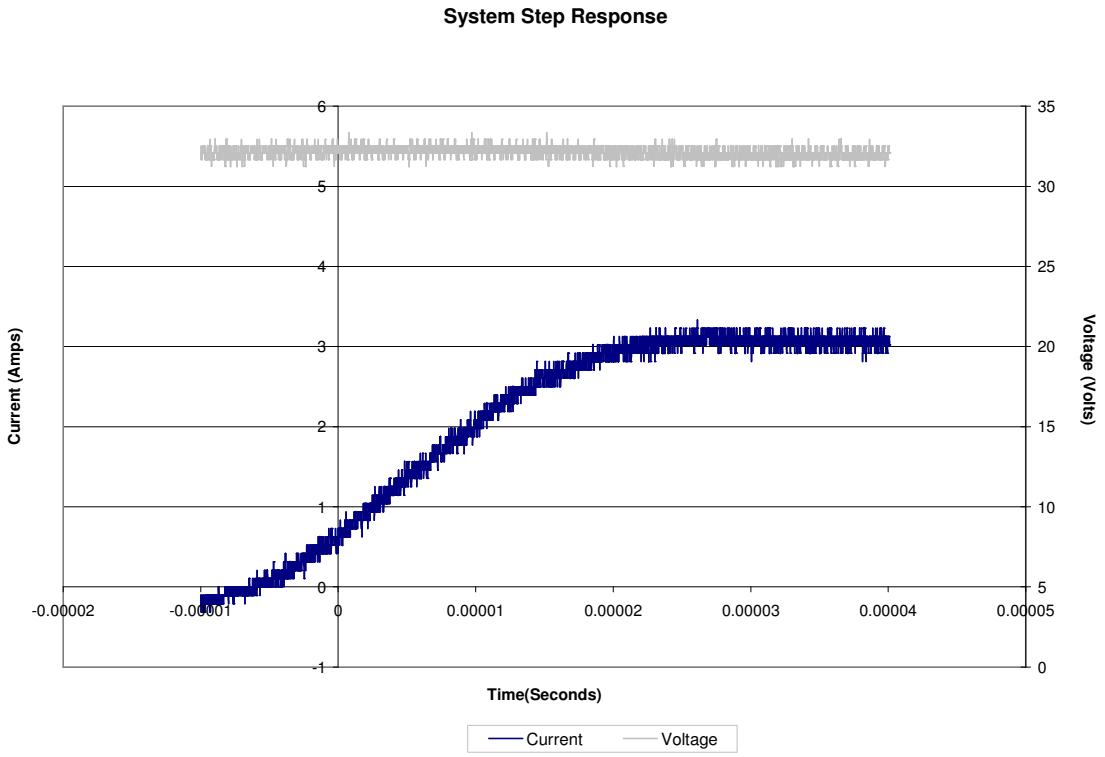


Figure 17: Step Response of the System With SSC set to Three Amps

Using the data obtained from testing the step response of the system, it can be seen in Figure 17 that the systems response is a simple first order response which follows the transfer function in Eq. (4.1) where the gain a is the current from the solar array string and the value of τ is set by the response time.

$$G(s) = \frac{a}{\tau s + 1} \tag{4.1}$$

The value for τ can then be calculated using Eq. (4.2) which uses the settling time T_s to calculate the value of τ.

$$\tau = \frac{T_s}{4} \tag{4.2}$$

Using Eq. (4.2) and a settling time of 30 microseconds from Figure 17, the value of τ was calculated to be 0.0000075 radians per second. Next this value was substituted into Eq. (4.1). Using this transfer function, the system was simulated in Simulink using Figure 18 in order to see if the step response of the system would match the response of the system in Figure 17. From simulating the system it was found that Simulink could not handle the speed of the response using the value calculated from the real systems response because there was a small oscillation introduced in the steady state of the simulation. In order to improve the model the value of τ was cut to 0.001 which was the fastest value which would simulate properly. However this appears acceptable for this system because the response time was found to be 0.008 seconds which is still plenty fast to simulate the response of the system. The final model for the plant which represents the transfer function of Eq. (4.1) can be seen in Figure 19 which includes a rounding block to represent the fact that only incremental numbers of channels can be turned on in the SASM.

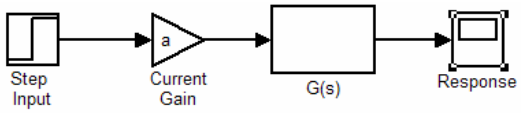


Figure 18: System Model

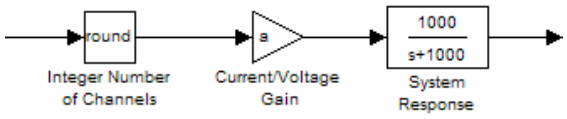


Figure 19: Final Plant Model

Having obtained the systems transfer function, the next step was to define the current gain *a* in the systems transfer function of Eq. (4.1). The current gain *a* is the current supplied from the solar array strings. The current from the solar array strings is a function of several different variables in the system. However before defining the current gain *a* several different assumptions need to made in describing the system. The first assumption which needs to be made is that the output current from all of the solar array strings is the same so that the transfer function of Eq. (4.1) can hold true for the response of the system. This is however a valid assumption to make because all of the solar array strings will be held to the same environmental conditions therefore leading to the fact that all of the solar array strings will have the same operating curve to define there operation. Using the assumption that all of the solar array strings will have the same operating curves, next it has to be assumed that all of the solar array strings have identical operating points. This is an acceptable assumption to make however because the battery is attached to the systems bus therefore the bus voltage will remain the same even when additional solar array strings are added or removed. As a result, if the voltage of the solar array strings remains the same then the output current from the solar array strings will also remain the same. With these assumptions in place to validate the transfer function describing the system, the next step is to define the operation of the solar array strings in order to define the current gain *a* in the systems transfer function.

In order to define the operation of the solar array strings, the first step was to characterize the output of the SAES which is used to simulate the solar array strings. The test which was preformed to characterize the SAES was a test where one solar array string was connected to a constant current load while the OCV and the SCC of the solar

array string were held constant. Then the constant current load was slowly increased while the output voltage of the solar array string was recorded. This procedure was repeated for several different OCV values and the results from these tests can be seen in Figure 20.

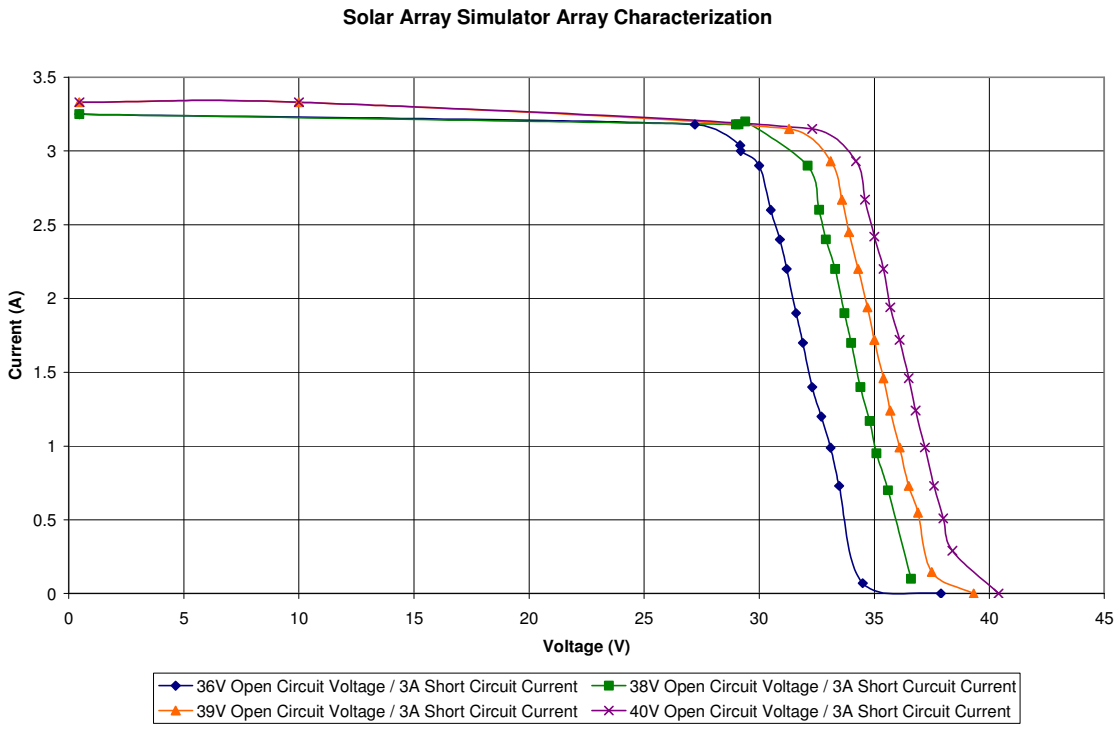


Figure 20: SAES Current Versus Voltage Curves

Next, the results obtained could be used to create a model which would define the current gain *a*, the current supplied by the solar array strings. The resulting model needed to define the solar array strings would need to be able to define the current from the solar array strings output current as a function of the bus voltage, the solar array strings OCV, and the solar array strings SCC. In order to generate the first part of this model, a look up table would be created which uses the test results of Figure 20 to define

the output current at three amps SCC as a function of the bus voltage and the OCV. The look up table to do this was generated by first picking the best and most representative curve from Figure 20 and the curve which was chosen was the curve with an OCV of 39 volts and a SCC of three amps. Using this curve, next identical curves were generated in one volt increments by shifting the chosen curve left and right. This was done because if the curves in Figure 19 are observed it can be seen that all of the curves are identical with the only difference being that the curves are shifted left or right depending on the OCV value. These identical curves in one volt increments create a lookup table which gives the solar array strings operating curve for any OCV at three amps SCC because the lookup table is capable of extrapolating value between the one volt increments in the lookup table.

The next step in describing the model for the solar array strings was to define the OCV and the SCC values over time for a LEO profile. The curves which were chosen to represent the LEO profile were the OCV and SCC curves in Figure 4 and Figure 5 respectively. These curves were chosen because they are the curves which are used to program the LEO in the SAES therefore they should be the exact values which are seen in the hardware. These profiles were added to the model using a lookup table which generates the OCV and SCC as a function of time.

With the OCV values defined over time, these values can then be feed directly into the model which was defined previously for the solar array gain a as a function of the voltage and the OCV at three amps SCC. The final portion of this model which needs to be defined is the way in which to scale the output current from three amps SCC to any SCC value. In order to do this scaling, a simple scaling factor is created where the

desired SCC is divided by the nominal current of thee amps and then this resulting scaling factor is multiplied by the output current from the lookup table which defines the output current at three amps SCC as a function of the OCV and the current bus voltage. This scaling can be seen clearly in Figure 21. As a result, this completes the model of the solar arrays for the LEO profile such that there is a model to define the current gain *a* of the solar arrays as a function of the bus voltage for a LEO simulation. The completed model can be seen in Figure 21.

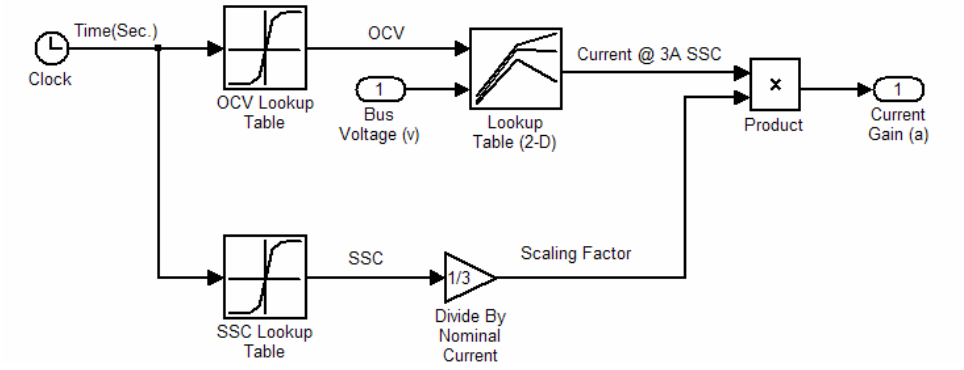


Figure 21: Model Defining the Current Gain a as a Function of bus Voltage

Having defined a basic model for the solar arrays, the next step was to develop a model for the battery which will help define the input Bus Voltage, *v*, to the solar array model in Figure 21. To begin the modeling process for the lithium ion battery some research was preformed on batteries and it was found that the best model to use for this simulation was a simplified model from [22]. The model which was used was slightly simpler then the model proposed by [22] because they propose two resistors in series with a voltage source and a capacitor around one of the resistors. The simplified model on the other hand ignores the capacitor and lumps the two resistances together. This is acceptable for this application because the total resistance of the actual battery is known

while the value of the capacitor, which is ignored, is not. The basic battery model which resulted can be seen in Figure 22. Also this model was used because it was a simple battery model which was proposed by several sources on battery modeling.

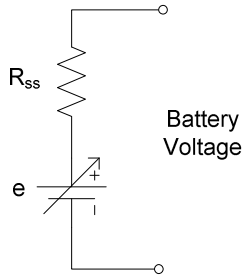


Figure 22: Battery Model

Knowing the basic structure of the battery two different parameters needed to be defined. The first parameter is the internal resistance of the battery. This parameter was a simple parameter to select because it was known that the internal resistance of the battery was found experimentally to be around 16 milliohms. The other portion of the battery model which needed to be defined was the variable voltage source as can be seen in Figure 22. The variable voltage source is a source where the voltage varies depending on the state of charge of the battery. From the paper on battery modeling it was shown that the voltage versus the state of charge is not very linear over the entire range of the state of charge versus voltage [22]. However, the assumption was made at this point that the relationship between the state of charge and the battery voltage would be linear. This would be an acceptable assumption as long as the state of charge of the battery did not fall below 50 percent because below this point the curve becomes very nonlinear while above this state of charge the linear assumption holds fairly true. Also for the test

currently being preformed the target lowest state of charge was found to be 90 percent therefore, the system will never fall into the nonlinear region for the state of charge as a function of voltage. Knowing this fact and the size of the battery, 60 amp hours, a simple equation could be determined which results in the battery voltage as a function of the current being supplied to the battery. The equations which were found to describe this relation ship can be seen in Eqs. (4.4) and (4.5) where *i(t)* is the battery current, SOC_{inital} is the initial state of charge of the battery, *w* is a gain based on the properties of the battery, and *b* is based on the properties of the battery.

$$SOC(i) = \left[\int i(t)dt + SOC_{inital}\right]\frac{1}{60} \tag{4.4}$$

$$v(i) = w * SOC(i) + b \tag{4.5}$$

In order to set the parameters for equation 4.4 the initial state of charge, $SOC_{initial}$, was set to 90 percent of 60 amp hours. The values for the parameters *w* and *b* in Eq. (4.5) were set using the method of trial and error when a constant 10 amp current was supplied to the battery model. The target to match in determining these values through a trial and error approach was to match the results obtained for the intuitive controller in Figure 16. In the end these values were tuned such that *w* was 9.95 and *b* was found to be 27.76. With the battery model completed, the complete model of the system could then be put together.

The complete model for the system can be seen in Figure 23. The model seen in Figure 23 incorporates all of the components previously described plus some additional loads which will be present in the real system. The additional loads which were not mentioned as can be seen in Figure 23 are the cable losses which were set so that there

were zero losses, the constant load which will be present in the system and set to three amps, and a variable load which can be switched on and off and any time. With the model completed, the next step is to verify the system model.

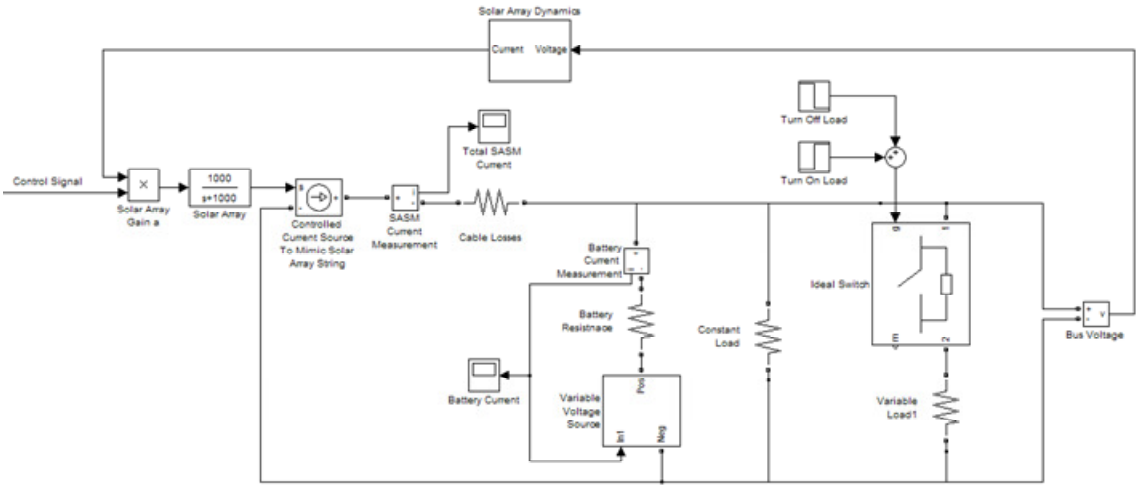


Figure 23: SASM Plant Simulink Model

4.2 Model Verification

The final step in the process of creating a model for the system was to verify the model to show that it matches the simulation results obtained. This test was done because if it proved successful, for the system model, it could be said that the system model matches the real system and the simulation model created can be used to verify future controllers developed prior to implementing them in hardware. In order to verify the model, the intuitive controller was implemented in a simulation which included the model created for the system in Figure 23. The resulting model which was created can be seen in Figure 24 which shows the Simulink model used to simulate the system for a

LEO test. The test results which were obtained for a LEO simulation can be seen in Figure 25.

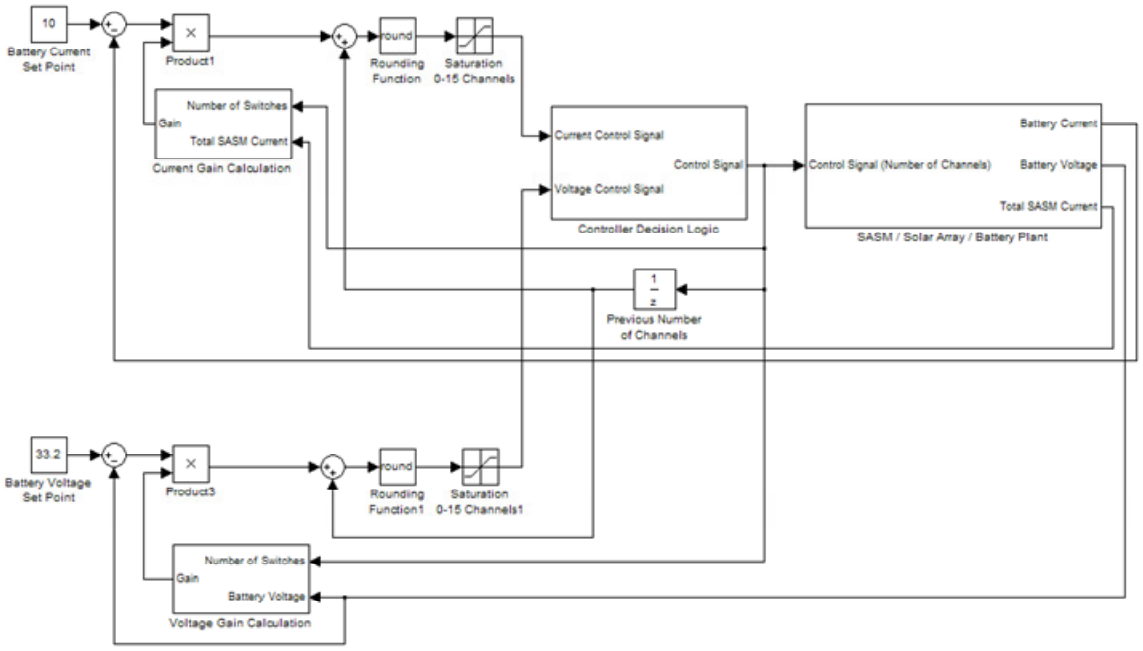


Figure 24: Intuitive Controller System Model

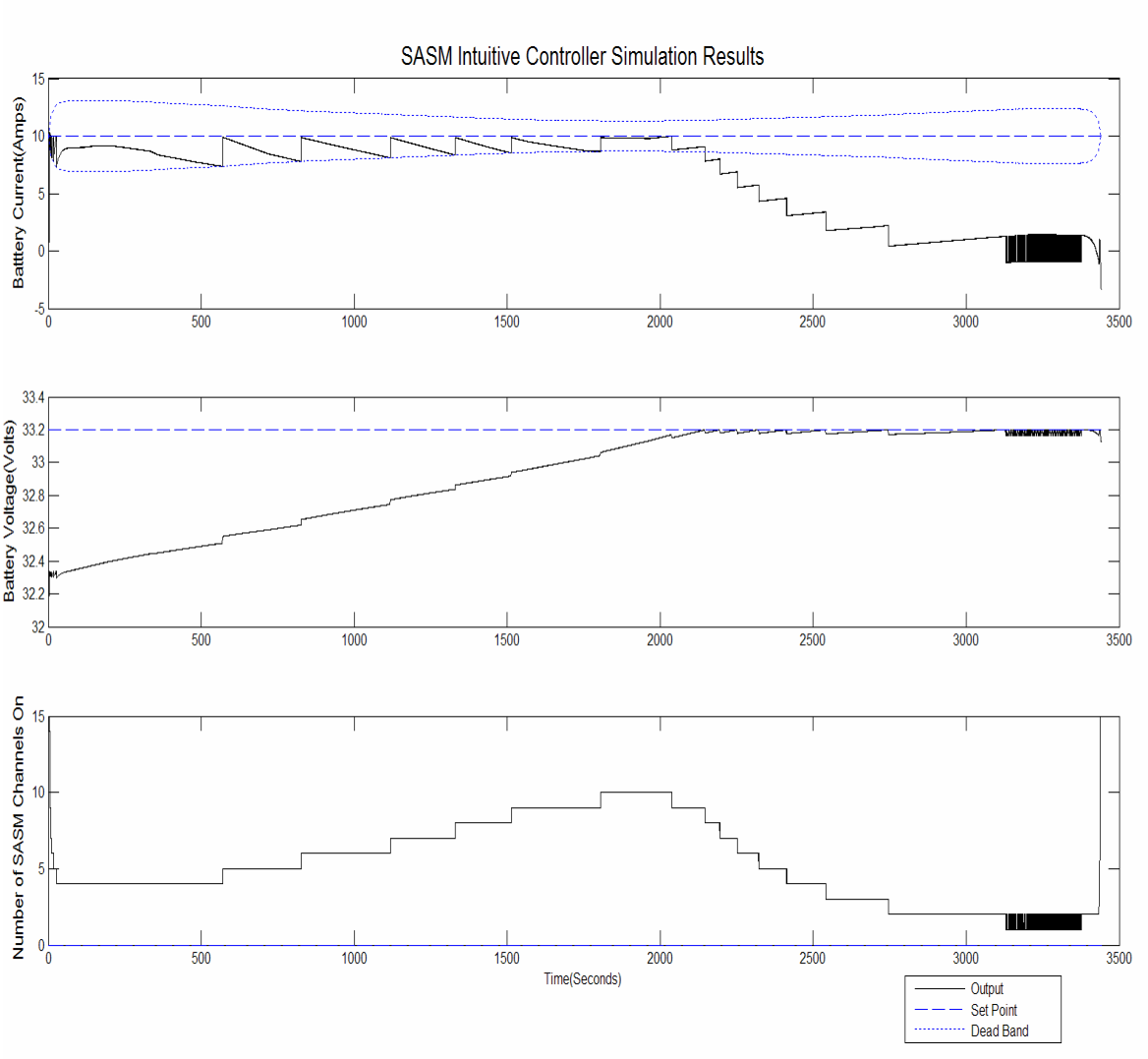


Figure 25: Model Validation Results

Overall the simulation results appear to match the results from the hardware test. The similarities in these results can be seen if the simulation results in Figure 25 are compared with the hardware test results of Figure 16. The only minor differences in the simulation appear at the very beginning of the simulation and at the very end of the simulation where there appears to be some excessive switching. However these minor differences are most likely due to the simplification of the battery model because in the simplification process some of the dynamics of the battery were left out therefore

probably making the battery model more linear then the actual battery. As a result this causes there to be a slight differences in the results obtained using the simulation. But overall, the simulation model is good enough to use in this application to test future controllers and have a fair level of certainty that the controller will work in hardware if it simulates properly.

CHAPTER V

IMPROVING THE CONTROLLER DESIGN

Having a better knowledge of the system from creating a system model and determining a transfer function for the plant, the next step was to try and improve the controller for the SASM. In trying to improve the SASM's controller, the first step was to complete a proportional integral (PI) control design for the system. After doing this, a PII controller design was investigated which lead to the development of a novel implementation of an integrator which was found to be able to control the system and remove excessive oscillation from the system when setup in conjunction with a PI controller. The development of this novel integrator implementation with a PI controller lead to a controller which was found to be effective in controlling the SASM as this will be shown through simulation and hardware test results.

5.1 PI and PII Controller Design

Having the plant defined with a transfer function, the next step was to try and design a controller for the SASM using a more defined and proven controls technique. The first controller which was studied for the SASM was a simple PI controller. In order to design the PI controller, first it was assumed that the discontinuities of the controller were removed to create a continuous time first order system as can be seen in Figure 26. Next the PI controller was designed using the method of pole placement. In order to use the method of pole placement, the systems transfer function was first derived as can be seen in Eq. (5.1) where a is the magnitude of the current, K_i is the integral gain of the controller and K_p is the proportional gain of the controller.

$$TF_{PI} = \frac{1000a * K_p * s + 1000a * K_i}{s^2 + (1000 + 1000a * K_p)s + 1000a * K_i} \tag{5.1}$$

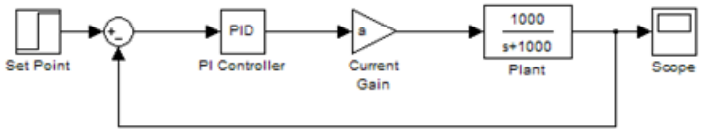


Figure 26: PI Control Loop

Having derived the transfer function of the system, the roots and poles of the system could then be calculated and this was done using a simple m-file in Matlab which used the roots command to find the poles and zeros of the transfer function. This Matlab file can be seen in Appendix B. In order to calculate the poles and zeros the current gain a was first assumed to be three, the maximum solar array current. From the transfer function of Eq. (5.1) it can be seen that there will be one dominant pole in the transfer

function because of the fact that the transfer function has one zero in the numerator and two poles in the denominator. As a result one pole and one zero can be placed in the same location to cancel each other out while the remaining pole can be set to control the speed of the response. In order to place the pole, the ideal dominant pole location was determined. The desired pole location is based on the fact that the desired response time of the system is 0.5 seconds and using the desired response time the approximate pole location can be calculated using Eq. (5.2) assuming that the damping coefficient is one.

$$\omega_n = \frac{4}{Setteling\ Time} \tag{5.2}$$

From Eq. (5.2), the desired pole location was found to be -8 rad/sec on the real axis. Different values of K_i and K_p were then tested for this system to try and achieve the desired pole placement location while holding the gain a, the solar array string current, constant. Having placed the dominant pole at the desired location while placing the other pole and zero to cancel each other out, the system was then simulated using the simulation of Figure 26 and the controller gains were fine tuned through simulation to achieve the desired response time. In doing so, it was found that best location to place the dominant pole was around negative nine on the real axis.

The PI controller then had the discontinuities of the rounding function and the zero order hold added into the system as can be seen in Figure 27. It was discovered through simulation that when the discontinuities were added into the system, the system response was not adequate and the system had to be retuned. After retuning the system to achieve the best response possible better then 0.5 seconds, it was found that the ideal dominant pole location using the ideal transfer function of Eq. (5.1) was around negative

three on the real axis. At this point the controller gains were .001 for K_p and 1 for K_i. Having tuned the system for a current gain a of three, next the value of the current gain a was adjusted and it was found that this gain has a direct effect on the response of the system. It was found in simulation that if the gain a is cut roughly in half to around 1.5, the response time of the system becomes 2.1 seconds as can be seen in Figure 28. As a result using a PI controller does not appear to be an effective option for this system because the PI controller is not capable of handling the wide ranging dynamics that occur in this system. Also it was noted in the simulations that if the current gain a was not an exact multiple of the set point the system introduced an undesired oscillation around the set point in the steady state as can be seen in the simulation result of Figure 28 where a is set to just below three. From the simulation results it can be seen that this system keeps incrementing and decrementing the control signal by the value of one. As a result this oscillation would be equivalent to the SASM continuously turning one switch on and off and it would be likely to occur all of the time because in the real world the SASM current will never be an exact multiple of the current. As a result, this oscillation shows that the PI controller does not meet the control goals for the SASM, thus showing that the PI controller is not an ideal candidate for this system.

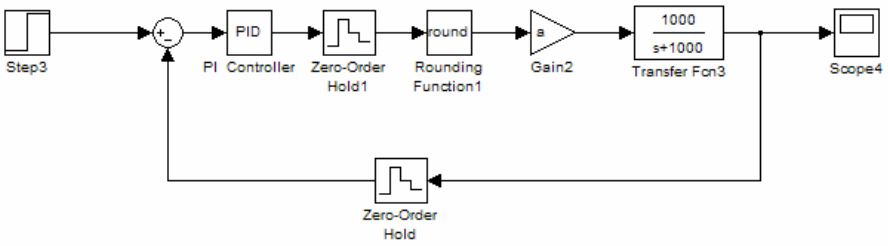


Figure 27: PI Control Loop With Discontinuities

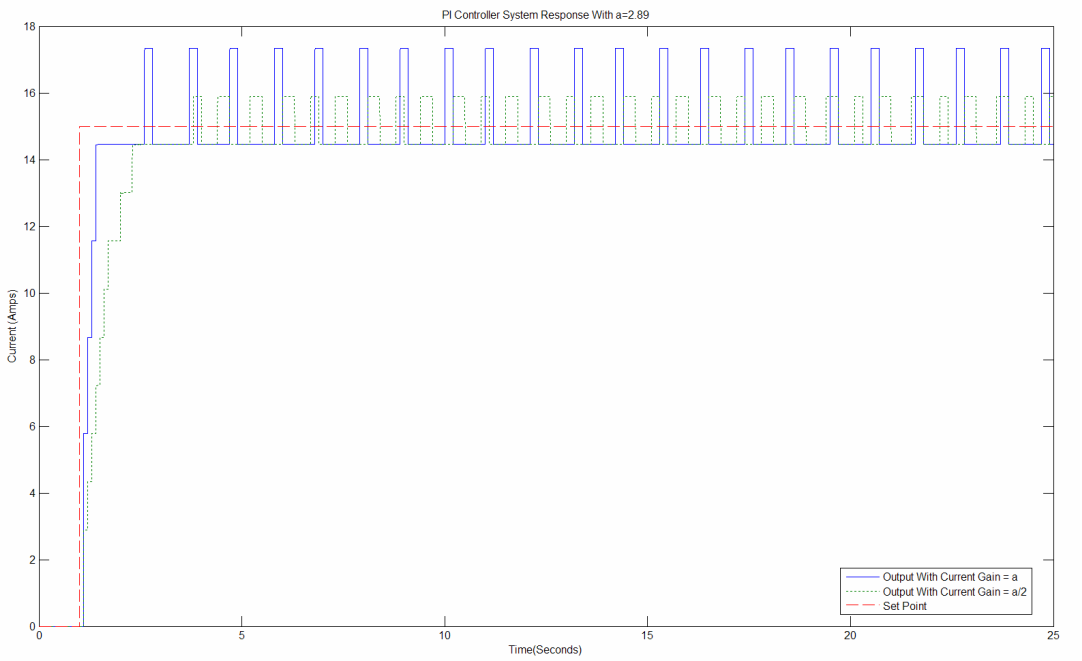


Figure 28: PI Controller Simulation Results of Figure 27

Using the PI controller as a basis for design, next the possibility of including an integrator at the output of the PI controller to create a PII controller was investigated. The PII controller was investigated because if the intuitive controller is studied it appears as thought the incremental structure is similar to a simple integrator so it was believed that potentially the intuitive controller could be thought of as a proportional controller with an integrator at its output. The possibility of using a PII controller was investigated in the same manner as with the PI controller. First the systems transfer function was derived from the systems model without discontinuities as can be seen in Figure 29.

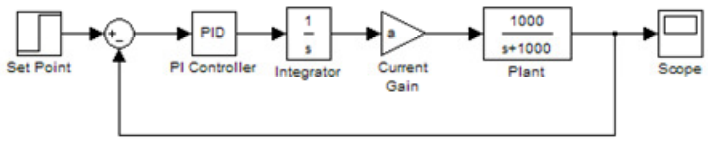


Figure 29: PII System Model

The transfer function was then derived for the system ignoring the discontinuities and the zero order holds of the system to create a continuous time system which can be analyzed using pole placement techniques. The resulting transfer function of the ideal continuous time system can be seen in Eq. (5.3).

$$TF_{PII} = \frac{1000a * K_p * s + 1000a * K_i}{s^3 + 1000s^2 + 1000a * K_p * s + 1000a * K_i} \tag{5.3}$$

Matlab was then used to compute the poles and zeros of the transfer function as can be seen in Appendix B. From computing the poles and zeros and using the method of pole placement, the system was tuned by adjusting K_i and K_p to have a single dominant pole at negative nine on the real axis because this was the pole location which achieved the desired response for the PI controller design. After tuning the system to achieve this desired pole location, it was found through simulation that the continuous time system without disturbances in Figure 29 could reach the set point in one half second if the dominant pole was placed at approximately negative nine. This tuning however did not satisfy the performance goal of the system when the rounding and zero order holds were added to the system as can be seen in the model of Figure 30. As a result the system was retuned using the simulation to achieve the best performance possible which is better then the desired response time of 0.5 seconds. The resulting response of the system can be seen in Figure 31. After tuning the system with discontinuities then the pole locations for the ideal situation were found such that the dominant pole was at negative three on the real axis which is the same as was found in the case of the PI controller. This result makes sense because the PI and the PII controller both contain one dominant pole which controls the response of the system. From simulating the system it was also found for the

PII controller that there was oscillation in the steady state when *a* is not a multiple of the set point as can be seen in the simulation results of Figure 31. Thus showing that the PII controller would not be a good choice for the SASM. As a result, the structure that appears similar to an integrator in the intuitive controller is not really an integrator based on the fact that the introduction of the integrator at the output of the system did not help to remove the steady state oscillations from the system.

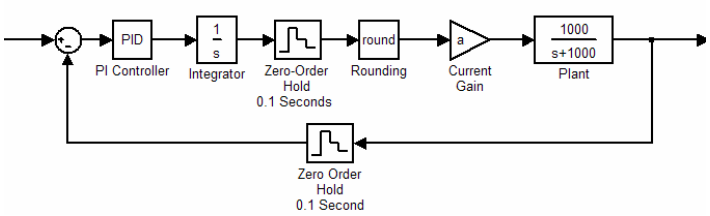


Figure 30: PII System Model With Discontinuities

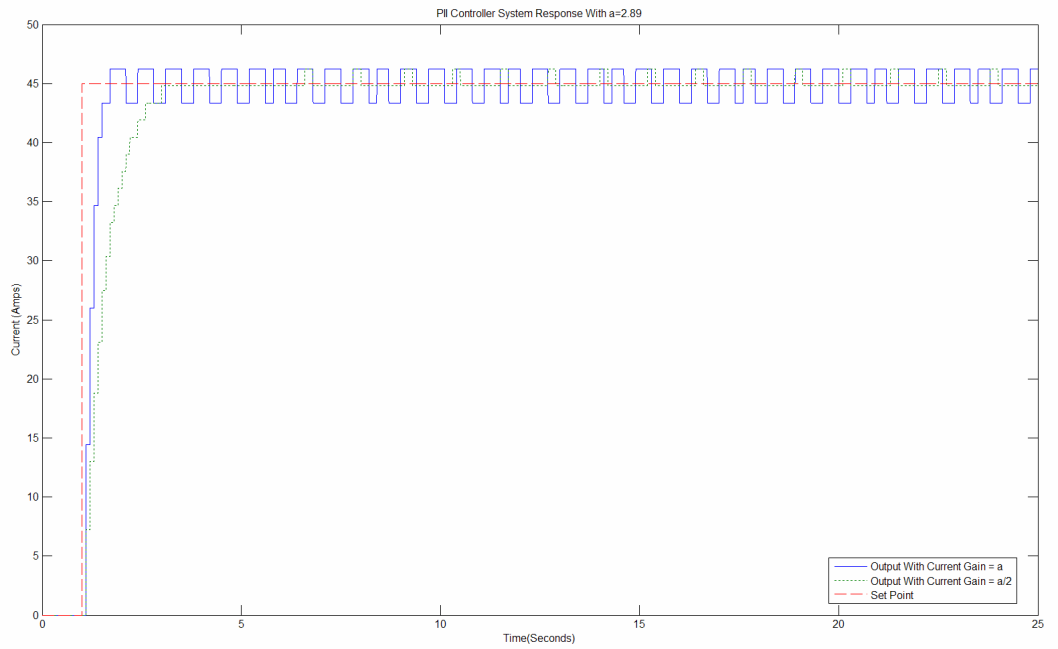


Figure 31: PII Simulation Results With Discontinuities

5.2 PI Controller with a Novel Integrator Implementation

After showing that the simple PI and PII controllers were not suitable for this system, the next step was to try and develop a controller that would be capable of meeting the system response time requirements while at the same time damping oscillations in the steady state. In order to try and remove the steady state oscillations from the system it is proposed at a novel integrator implementation be applied to the output of a PI controller. The integrator structure which is proposed can be seen in Figure 32.

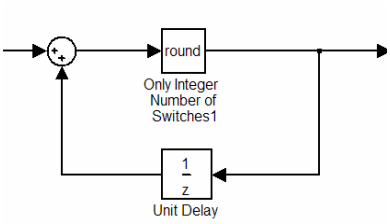


Figure 32: Novel Integrator Structure

To begin the system design with the novel integrator structure, first the model for the simple current control case was built as can be seen in Figure 33. This overall control structure with a PI controller and the novel integrator at its output will be for simplicity sake referred to from this point forward as an incremental PI controller. After setting up the model, the current gain a was set for the maximum of three amps and the system was tuned using the values of K_p and K_i found in tuning the PII control system as a starting point. No real tuning method such as pole placement could be used for this controller because no proven method is known to exist. The incremental PI control structure was then tuned using the simulation of Figure 33 and it was tuned such that the best response

time of the system was found when the current gain *a* was slightly less then three. The response of the system which is better then the desired half of a second can be seen in Figure 34. This response in Figure 34 is important because in both the PI and the PII controllers the system was found to oscillate in the steady state when the gain *a* was not an exact multiple of the set point as will be the case in the real system. However as it can be seen in the response of Figure 34 and Figure 35 which is the same simulation but longer, the system is shown not oscillate excessively as was found for the PI and the PII controllers. Next the gain *a* was cut in half to observe the response of the system. At this point it was noted that the systems response was still acceptable because the response time became around half a second for an input step change as can be seen in Figure 34 along with this the response was also found acceptable in the steady state as can be seen in Figure 35. This shows that the incremental PI controller is capable of operating over a wide range of operating conditions as will occur in the real system without producing excessive oscillations in the steady state. The oscillations which did occur in the steady state did however appear to be very controlled such that they are very slow and this results in an averaging effect around the set point which is desirable for this application.

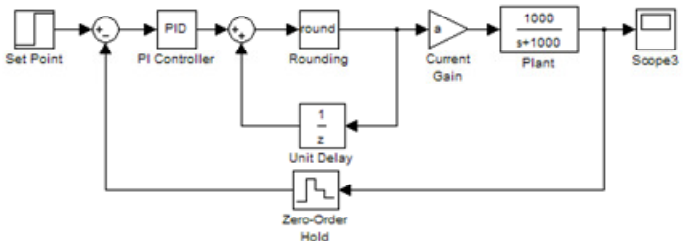


Figure 33: Incremental PI Control Structure

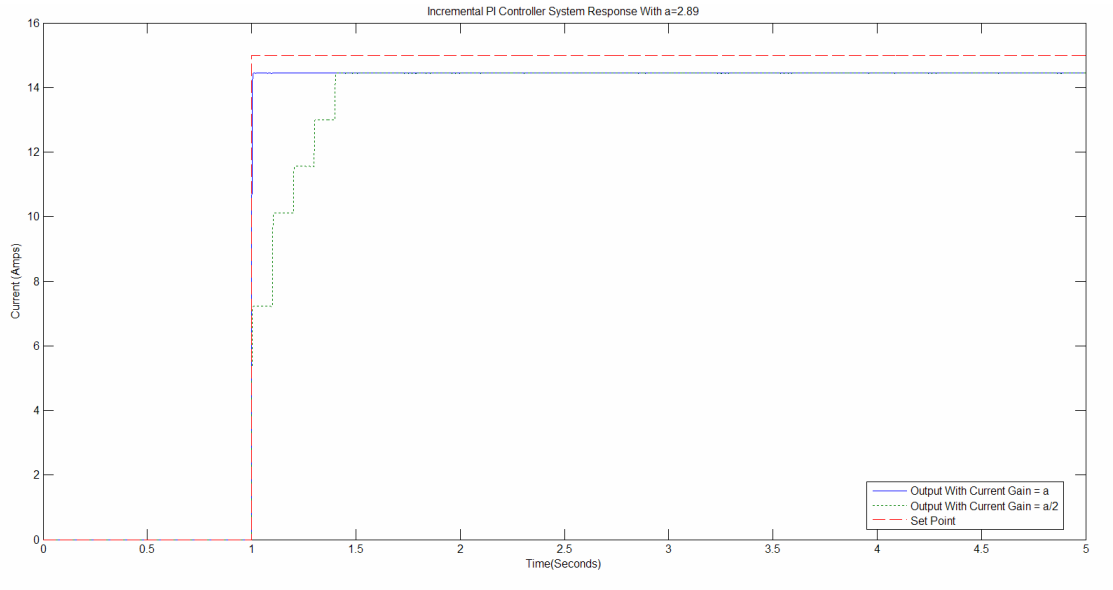


Figure 34: Incremental PI Controller Transient Response

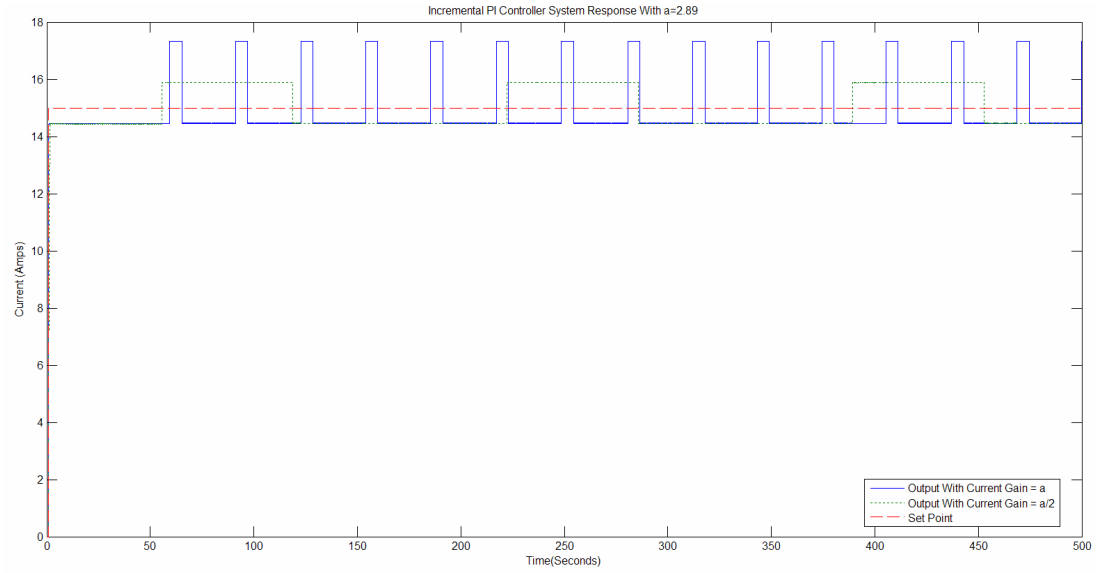


Figure 35: Incremental PI Controller Response Over Time

Having shown that the simple simulation of Figure 33 was acceptable over a wide range of values for the gain *a*, the next step was to incorporate the incremental PI controller into a full LEO system simulation as was done for the intuitive controller to show the effectiveness of the controller over a wide range of dynamics. To do this, the

incremental PI control would need to be extended from the current control mode into the voltage control mode.

To design the controller for the voltage control mode, this was a much simpler task because the dynamics of the voltage control mode are not nearly as wide ranging as those of the current control mode. As a result, the voltage control mode was designed using a structure identical to the current control mode. Having designed the basic structure, the controller was then incorporated into the full simulation as can be seen in Figure 36 where the voltage control mode would be later tuned using the full simulation. One other issue with implementing the full simulation was designing the rules for when to change between the two different control modes. The rule base for the incremental PI controller is slightly different then the rule base used in the intuitive control. The rule base which was created looks at several different inputs to determine which control mode to go into. The rule base for selecting the control mode can be seen in the flow chart of Figure 37 which is incorporated in the full simulation of Figure 36 as the control mode decision block.

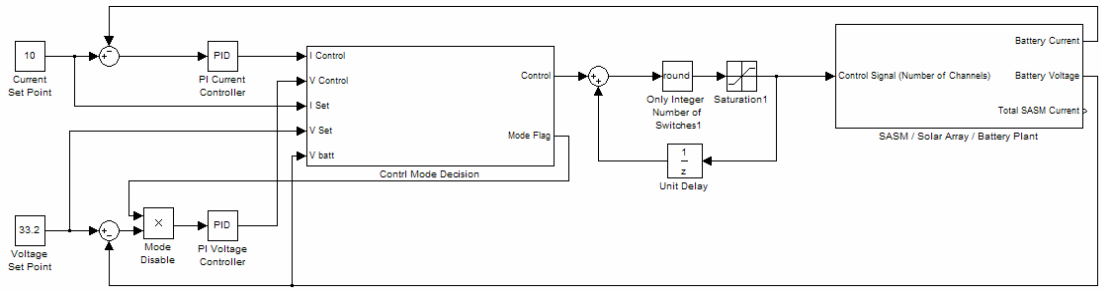


Figure 36: Incremental PI Controller Full Simulation

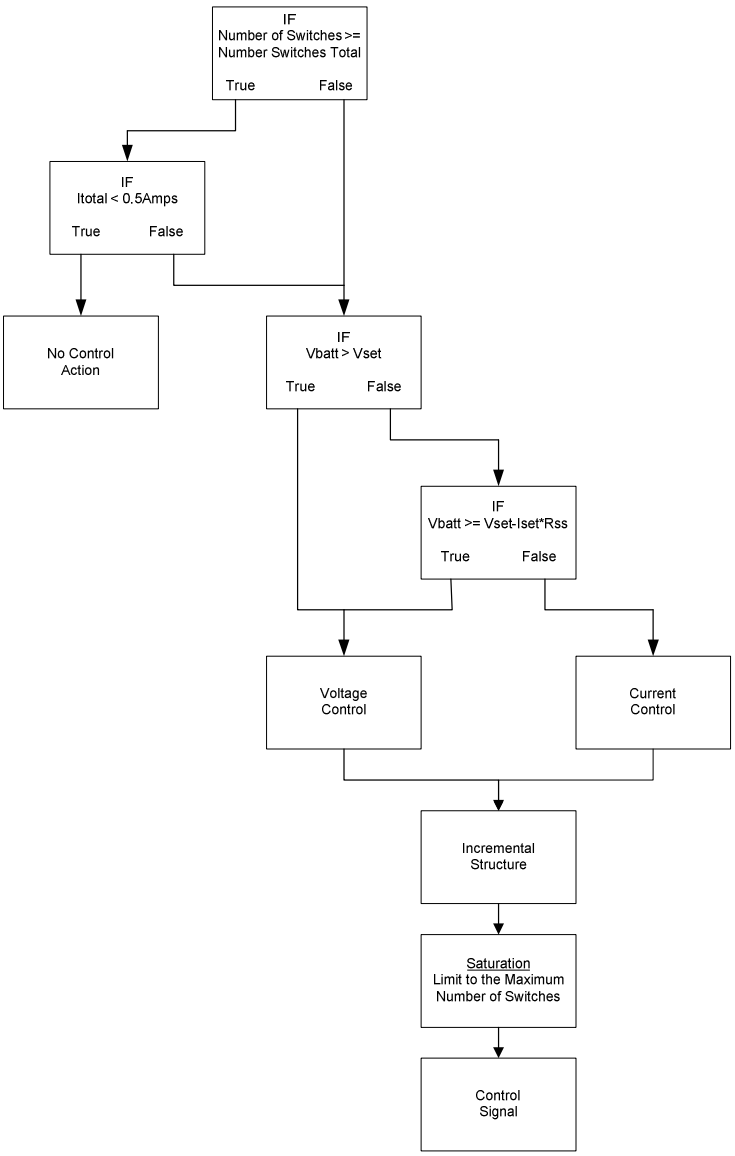


Figure 37: Incremental PI Controller Mode Selection

Having created the full simulation for the system, the next step was to run the simulation to see how the incremental PI controller in the SASM would handle the LEO. From running the full simulation the resulting outputs for the battery current and battery voltage were obtained as can be seen in Figure 38. The simulation results show that the system appears to follow the desired charging profile for the lithium ion battery which can be seen in Figure 6. The simulation results also show that the output is a smooth saw

tooth type of a waveform centered around the set points for both the voltage and the current. This shows that the incremental PI controller creates a smooth output and that the outputs average value should come close to matching the desired set point for either the current or the voltage. This appears to be better then the intuitive control in simulation because in the intuitive controller the current sits bellow the set point within the defined dead band for the system. One other advantage which can be seen in the simulation over the incremental controller's simulation is that in the voltage control mode at the end of the charging cycle, there appears to be less oscillation then in the simulation for the intuitive controller therefore showing another potential benefit of this controller in simulation. Therefore after simulating the incremental PI controller, it appears as though this controller should be implemented in the SASM in order to study its real world effectiveness.

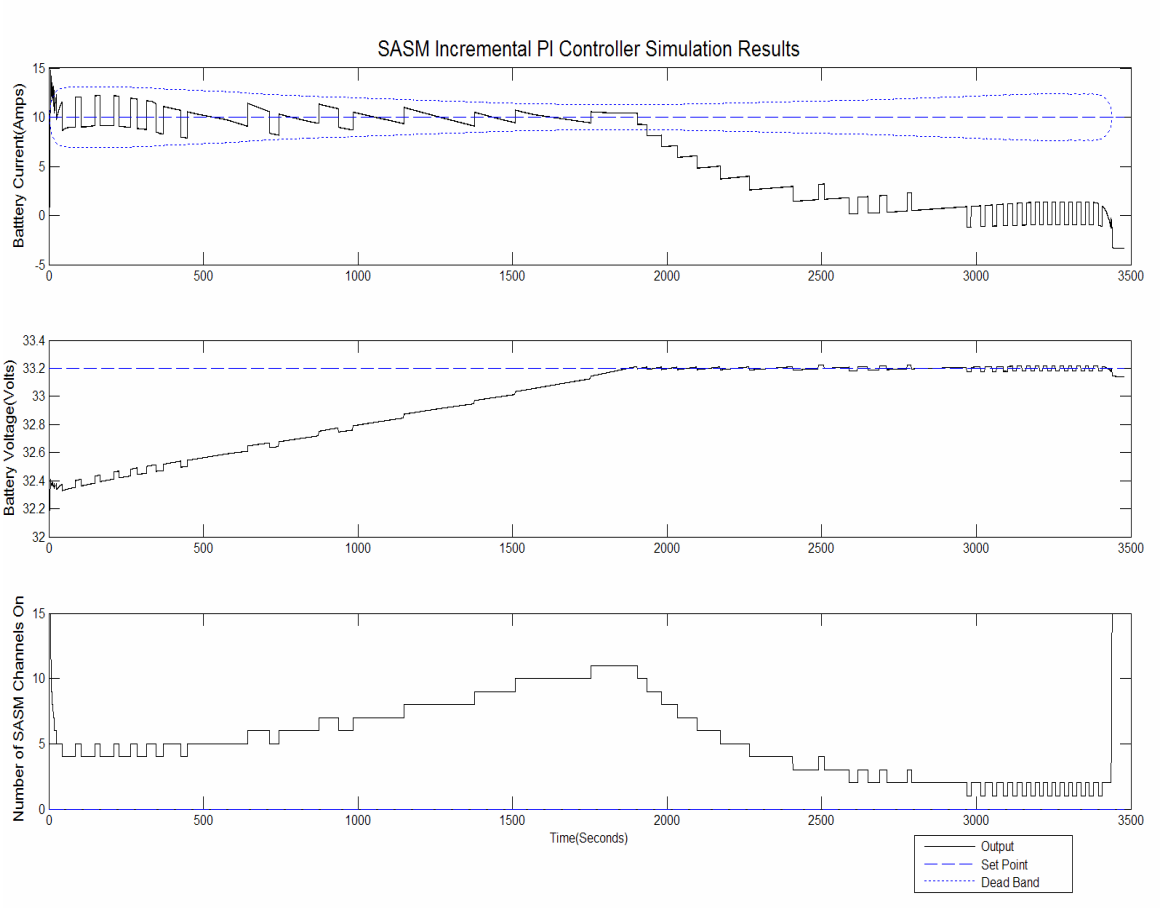


Figure 38: Incremental PI Control Simulation Results

5.3 Incremental PI Controller Analysis

Before going any further with the incremental PI controller, this controller needs to be analyzed to see why this controller behaves the way it does. The portion of this controller which needs to be analyzed first is the novel integrator implementaiton which can be seen in the block diagram of the incremental PI controller in Figure 39. At first glance this structure appears to resemble the structure of an integrator implemented in

discrete form. However if the input output equation for the novel integrator implementation is written it can be seen that it does not match the equation for discrete integrator. The equation for the novel integrator structure can be seen in Eq. (5.4) where the input is $u_1(k)$, the output is $u(k)$, and the round function is $r(\cdot)$.

$$u(k) = r(u_1(k) + u(k-1)) \tag{5.4}$$

Next this equation can be compared to the equation for the implementation of a discrete integrator which can be seen in Eq. (5.5) where the input is $y(k)$, the output is $u(k)$, and T is the sampling time.

$$y(k) = y(k-1) + T * u(K) \tag{5.5}$$

As a result it can be seen that these two equations are not the same thus proving that the novel integrator proposed is not a true integrator rather it is something else.

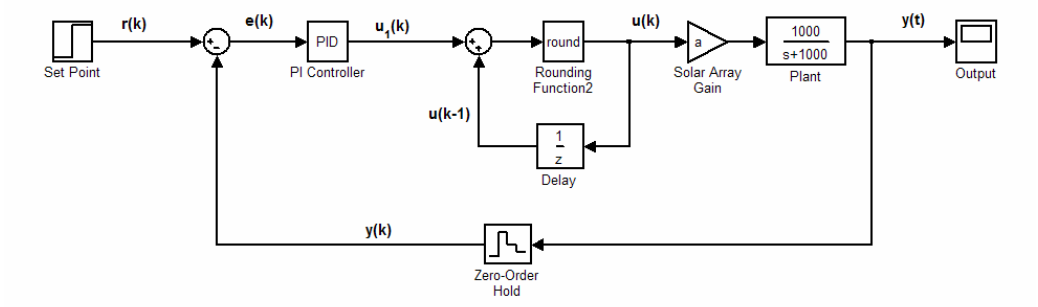


Figure 39: Incremental PI Controller Block Diagram

In order to further analyze the system, the system was simplified by removing the rounding from block from the system as can be seen in Figure 40. After doing this, the equations describing the system can be written according to Eq. (5.6) where $u_1(k)$ is the input, and $u(k)$ is the output.

$$u(k) = u_1(k) + u(k-1) \tag{5.6}$$

From Eq. (5.6) it can be seen that the incremental structure without the rounding function is similar to the discrete integrator in Eq. (5.5). In order to make these equations match it can be seen that if the input to the integrator in Eq. (5.5), $u_1(k)$, has a factor of the inverse of T factored into it the two equations, Eq. (5.5) and Eq. (5.6), will match. As a result to get the ideal integrator of Eq. (5.5) to match the block diagram of Figure 40, a factor of the inverse of T will be introduced into the system as can be seen in Figure 41, thus showing the relationship between the integrator and the proposed structure if the rounding function is removed. However as a result it can be seen that the rounding block changes the overall operation of the simple integrator implementation.

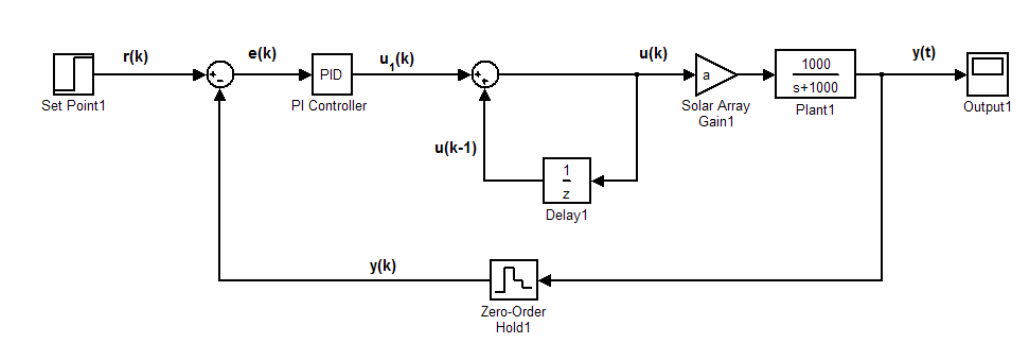


Figure 40: Incremental PI Controller Without the Rounding Function

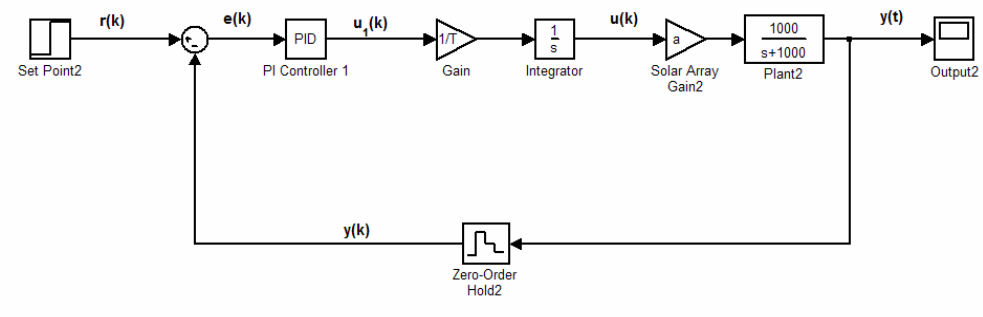


Figure 41: Continuous Time Equivalent to Figure 40

After simplifying the system and finding the continuous time equivalent of the reduced system without the rounding function as can be seen in Figure 41, the next step was to go back to the original equation of the proposed integrator structure which can be seen in Eq. (5.4). From this equation it does not appear as thought there is a simple way to mathematically explain the way in which the nonlinear rounding function, $r(\cdot)$, affects the integrator structure because the rounding function introduces a discontinuity into the system. To try and understand this structure in the simplest way and to try and explain its operation is to look at it logically. It appears that the rounding function acts as a damper, because it requires the signal from the PI controller, $u_1(k)$, to be large enough to move an entire channel before anything happens therefore showing that the round function acts as a damper by introducing what can be thought of as a dead zone in the system.

The modified integrator structure is an important part of this controller because it does most of the work of the controller. The fact that this structure does most of the work can be seen from the fact that every iteration this structure introduces the previous control signal into the new control signal thus showing that it is responsible for generating a large portion of $u(k)$. As a result of this structure maintaining the previous control signal in the current control signal, this leaves the PI controller to be tuned differently then it can be tuned as a PI or PII controller as was previously analyzed. From studying the system in simulations, it was found that the proportional part of the PI controller is tuned to due most of the work in moving the system during the transient response or another way of looking at it is the proportional gain sets the transient response. As a result the proportional gain can be tuned to achieve the desired response of the system during transients. Also it was observed that if this system is compared to the PII controller the

proportional gain of the incremental PI controller could be tuned such that its gain is less then the gain used in the PII controller. This means that the proposed integrator structure in the controller allows the proportional gain in the controller to be detuned while at the same time achieving the desired response from the system. This in the end means that the proportional portion of the PI controller in the incremental PI controller can be tuned to achieve good performance while at the same time avoiding excessive oscillation.

The integral control action from the integrator in the incremental PI controller is the remaining portion of the controller which needs to be studied. From studying the system in simulations it was found that integrator gain can be tuned to control the oscillation rate of the system such that the higher the integral gain is the faster the oscillation rate occurs. This appears to make sense because the main purpose of adding the integral action in the controller is to remove the steady state error. Therefore the higher the gain is the faster the error will accumulate which will in turn lead to faster oscillation around the set point because the system will never be able to achieve the set point due to the course nature of the SASM. Even though this novel integrator implementation can not be explained mathematically due to the fact that the rounding function is discontinuous, the response of this controller appears to make sense if it is studied logically.

5.4 Hardware Implementation

Having completed the simulations for the incremental PI controller and showing that the incremental PI control system for the SASM works in simulation, the next step was to show that the controller works in hardware. To implement the incremental PI controller in hardware, the same basic code which was used for the intuitive controller was modified so that the controller was the incremental PI controller rather then the intuitive controller. A description of the basic microcontroller code can be seen in Figure 14. Using this as the basic code structure the first step was to discretize the PI controller. The discretitization of the PI portion of the controller was completed in two parts. The proportional part of the controller is simple to discretize because it simply the input which in this case is the error multiplied by a gain. The discretization of the integral is slightly more difficult. In order to discretize the integrator portion of the PI control, the forward Euler approximation method was applied [23]. The forward Euler method is described in Eq. (5.7) where u is the input, y is the output, K_i is the integral gain and T is the sample time.

$$y(n) = y(n-1) + Ki * T * u(n-1) \tag{5.7}$$

After discretizing the controller, the next step was to code up the control algorithm for the incremental PI controller which was a simple process once the system was discretized. Once this was completed, the new controller for the SASM was loaded into the SASM microcontrollers and the system was tested. The first test which was preformed on the system was the same continuous LEO test which was preformed for the intuitive controller. This test was preformed to show that the controller could match the

desired charging profile for the lithium ion battery without disturbances being applied to the system. This test was completed successfully as can be seen in Figure 42 which shows one of the charging cycles which is representative of all of the charging cycles which were completed. After completing the LEO test with disturbances applied to the output of the system, the next step was to apply load changes at the output of the system. The load changes which were applied to the system were the same load changes which were applied to the intuitive controller. The results which were obtained from viewing the response on the oscilloscope can be seen in Table 1 which compares the disturbance rejection test results for the two different controllers side by side.

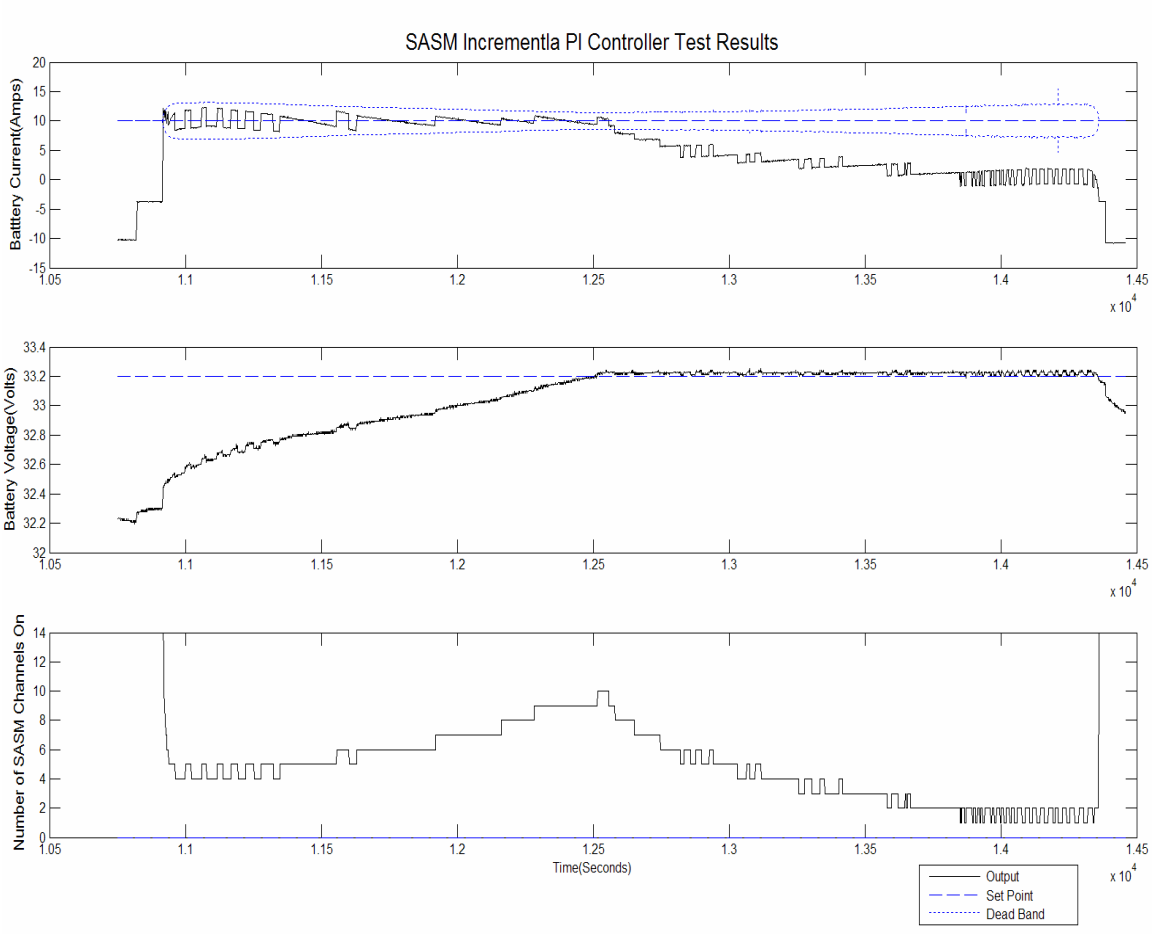


Figure 42: Incremental PI Controller LEO Test Results

From studying the test results it appears as though the incremental PI controller is capable of charging the lithium ion battery while at the same time rejecting disturbances. The incremental PI controller appears as though it is capable of properly charging the lithium ion battery because if the desired charging profile of Figure 6 is compared with the results in Figure 42 the results obtained from the test results appear to follow the desired charging profile. The only difference is the saw tooth looking charging pattern which occurs because of the fact that only single channels can be turned on and off in the SASM. Also an interesting comparison to make is between the simulation results in Figure 35 and the actual test results in Figure 42 because the actual test results appear to match the simulation results fairly closely. One other observation which can be seen in the test results is the fact that the resulting outputs appear to ride around the set point, thus showing that a true average output equal to the set point value is obtained.

The next test results which can be analyzed are the disturbance rejection, or load change, test results which can be seen in Table 1. From these results it appears as thought the average response time in the constant current control mode is 314.2 milliseconds which is well below the desired response time of 500 milliseconds and the results show that no response time went above the 500 millisecond target. The voltage control mode response shows a somewhat different story. The average response time was 567.5 milliseconds which is fairly close to the desired response time of 500 milliseconds for the system. However, even thought the average response time is slightly longer then the desired response time the system still performs good enough to be acceptable because the average response time shows that the system is within one control cycle of meeting the target. One good reason for this may be the fact that the system has not been tuned

perfectly yet due to the fact that it takes 90 minutes to complete one cycle and only a limited number of cycles were completed and the system was cautiously each cycle due to the fact that the controller tuning appeared to be in the ball park. The fact that the incremental PI controller did not meet the desired performance specifications in the voltage control mode will be discussed in more detail in the next section because the same problem of being one controller cycle too long was observed in the intuitive controller on several trials.

5.5 Controller Comparison

After designing and implementing the different controllers for the SASM, the intuitive controller and the incremental PI controller appeared to meet the goals set forth for the SASM controller in Section 2.5. From directly comparing the two controllers it appears as thought the incremental PI controller has the advantage when meeting the desired set points. The incremental PI controller has the advantage in this category because this controller results in a true average output around the set point. On the other hand, the intuitive controller does not have this averaging feature, rather the intuitive controller causes the system to sit generally below the set point with in the dead band region which was defined as can be seen in the test results of Figure 16. Even thought it is acceptable for the controller to sit within the dead band the incremental PI controller still has the advantage because it is capable of producing a true average output equal to the set point.

A second comparison which can be made between the intuitive controller and the incremental PI controller is in the controllers disturbance rejection capabilities or this can be thought of as the ability of the controller to react to load changes. The better controller at reacting to load changes appears to be the intuitive controller. This can be seen by analyzing the results in Table 1. From the results of Table 1 it can be seen that the average response time for the intuitive controller in both the voltage control mode and the current control mode is better then the incremental PI controller. Even thought the response from the incremental PI controller is slower in the current control mode, the incremental PI controllers response in the current control mode is still faster then the desired response time of 0.5 seconds in all cases. In the voltage control mode, the incremental PI controllers average response time is longer then the intuitive controllers response time and it should be noted that the incremental PI controllers average response time is slightly longer, 67.5 miliseconds longer then the desired response time. The voltage control mode results however appear to be acceptable because if all of the response times are studied for both controllers it can be seen that both controllers have some responses longer then the desired response time of 500 milliseconds.

On the surface it appears as though both controllers are unacceptable in the voltage control mode because they have some response times longer then the desired response time. However these longer response times appear acceptable because all of the responses are recovered within an additional controller cycle which is 125 milliseconds. In all actuality response times up to 750 milliseconds are only one controller cycle longer and this occurs because if the load change were to occur just after a controller cycle were to have finished this would give the worst case scenario which would still be only one

controller cycle too long. As a result this shows that it was not possible in the voltage control mode to achieve the desired performance with a control rate which is below the rule of thumb control rate but being one controller cycle too long appears acceptable for this application.

As an overall result it appears as though when the average response times are studied the intuitive controller appears to have the advantage. On the other hand if the results are analyzed more thoroughly both controllers appear to meet the performance requirements in the current control mode in that they never exceed the 500 millisecond mark. In the voltage control mode both controllers appear to have a number of trials which are one controller cycle too long. As a result the performance of both controllers could be considered equal in the voltage control mode. Overall both controllers appear to be capable of meeting or at least coming close to meeting the desired performance specifications which as was shown earlier is very difficult to achieve with a control rate below the rule of thumb control rate.

Overall the better controller between the intuitive controller and the incremental PI controller appears to be the incremental PI controller. This controller appears to be the better choice for the SASM for several different reasons. The first is that the steady state performance of the incremental PI controller gives a true average output around the set point. The second advantage of the incremental PI controller is that the incremental PI controller is probably more stable from the fact that it does not rely on gain calculations each iteration. This is a disadvantage of the intuitive controller because the controllers gain calculation relies on the fact that all of the channels are operating properly thus showing that if a number of solar array strings were to fail the gains may not be

calculated properly. As a result, the incremental PI controller is probably the better choice for the SASM even thought the intuitive controller appears to be an effective alternative to control the SASM.

CHAPTER VI

CONCLUSION

Overall this research takes the development of a SASM from a design concept to implementation in hardware that meets all specifications. The design process consisted of designing the SASM hardware, integrating the SASM with the other components in the system and then finally completing a controller analysis of the SASM. The controller analysis completed started with an intuitive controller, which then lead to the development of a novel implementation of an integrator which was proven to work in simulation and hardware test results. After completing this work with the SASM, there are several open doors which can be explored in future research.

6.1 Summary

This research dealt with the development of a solar array regulator for a space application. The regulator which was developed from concept to final hardware was a series array regulator called a SASM. This regulator operates in a manner such that the SASM turns solar array strings on or off depending on the demands of the system. The demands of the system are properly charging a lithium ion battery which is attached to the systems power bus while at the same time rejecting disturbances which may arise from unknown load changes in the system. In order to fully understand the operation of this system, first the system was thoroughly described so that the interaction among all of the components in relation to the SASM could be understood. Having a thorough understanding of the system, the next step was to define the control objectives for the SASM. The main control objective for this system was to charge the lithium ion battery to meet a given charging profile while at the same time satisfying the demands of the other unknown loads in the system. In addition the main objective of the SASM controller was to avoid excessive switching in the SASM while maintaining disturbance rejection capabilities.

With the goals for the controller in place, the system was first controlled using an intuitive controller because the exact dynamics of the system were not known due to the fact that the system was incomplete at this point in the SASM controllers development. After designing the intuitive controller, it was implemented in hardware to show the effectiveness of the intuitive controller and to learn about the system dynamics. After learning about the system through the intuitive controller and other system tests, the next

step was to develop a model for the system using the knowledge obtained. The system model which was developed combined a theoretical knowledge about the different devices in the system along test results to create an accurate model.

With a model created and an understanding of the system down to the level of a transfer function for the SASM a more complete controller analysis could be applied to the system. The controllers which were studied with this system were a PI and a PII controller. These controllers were both found to meet the control goals for the system as far as response time but they were found to oscillate in the steady state. In order to combat this steady state oscillation, a novel implementation of an integrator is proposed. This novel integrator in conjunction with a PI controller was found to adequately control the system over a wide range while at the same time removing excessive oscillation from the steady state. After showing that this novel implementation of an integrator worked to remove oscillation in the steady state, next an analysis of this proposed novel integrator implementation was performed and it was found that this structure is similar to an integrator but the addition of a rounding function changes the dynamics of the integrator structure completely.

After studying the proposed novel integrator structure, the next step was to apply the controller to the SASM in hardware. From performing hardware tests in the SASM with the incremental PI structure it was found that the controller was capable of controlling the system and effectively charging the lithium ion battery. After showing the effectiveness of this controller, the incremental PI controller was compared with the intuitive controller and it was found that the incremental PI controller would be more effective at controlling the SASM. As a result of this research, it was found that the

SASM could be effectively controlled and it was shown that a new novel implementation of an integrator is capable of removing oscillations in the steady state while operating over the wide ranging dynamics of the SASM.

6.2 Future Research

The development of the SASM controller leaves several open doors for research opportunities and room for hardware improvements. The main improvement which could be made is improving the redundancy of the SASM controller so that if one board were to fail the system would still operate. The ground work for this redundancy has already been laid but it would be a necessary improvement if the SASM were to be used in a real application. The area of research that this project could lead to in the future is a more thorough study of the incremental PI control structure. This could be an area of future research because it would be interesting to see if the novel integrator implementation could be used successfully in other applications which include dead zones or stiction to limit excessive oscillations in a system or if it could be extended to a more general frame work then the specific application which is presented here.

REFERENCES

1. Patel, Mukund R. Spacecraft Power Systems. CRC Press, 2005.

2. Wertz, James R., and Wiley J. Larson. Space Mission Analysis and Design 3rd ed. Microcsm Press, 1999.

3. Patil, A.R., B.H. Cho, D. Sable, and F.C. Lee, "Design and Test Hardware for a Solar Array Switching Unit," *27th Intersociety Energy Conversion Engineering Conference*, pp. 179-184, San Diego, CA, Aug. 3-7, 1992

4. Button, Robert M., "Development and Testing of a Source Subsystem for the Supporting Development PMAD DC Test bed," *26th Intersociety Energy Conversion Engineering Conference*, Boston, MA, Aug. 4-9, 1991.

5. Button, Robert M., "An Advanced Photovoltaic Array Regulator Module," NASA Technical Memorandum 107304, 1996.

6. Xiao, Weidong, William G. Dunford, Patrick R. Palmer, and Antoine Capel, "Regulation of Photovoltaic Voltage." *IEEE Transactions on Industrial Electronics*, Vol. 54, No. 3, 2007, pp. 1365-1374.

7. Usner, Eric P., and Michael M. D. Ross. "Recommended Practices for Charge Controllers," International Energy Agency Report IEA PVPS T3-05, 1998.

8. Sanidad, L., R. Parsons, Y. Baghzouz, and R. Boehm, "Effect of On/Off Charge Controller on Stand-Alone PV System Performance," *35th Intersociety Energy Conversion Engineering Conference and Exhibit*, pp. 1497-1501, 2000.

9. Larsson, Mats, Dragana H. Popovic, and David J. Hill, "Limit Cycles in Power Systems due to OLTC Deadbands and Load-Voltage Dynamics*," Electric Power Systems Research*, Vol. 47, 1998, pp. 181-188.

10. Horswill, Ian Douglas, "Simple Control Loops," Northwestern University, 1999.

11. Abdulgalil, F. and H. Siguerdidjane, "Nonlinear Friction Compensation Design For Suppressing Stick Slip Oscillations In Oil Well Drillstrings," International Federation of Automatic Control, 2004.

12. Papadopoulos, Evangelos G. and Georgios C. Chasparis, "Analysis and Model-based Control of Servomechanisms With Friction," *IEEE International Conference on Intelligent Robots and Systems*, Vol. 3, 2002, pp. 2109-2114..

13. Wu, Bing-Fei, Jau-Woei Perng, and Hung-I Chin, “Limit Cycle Analysis of Nonlinear Sampled-data Systems by Gain-Phase Margin Approach,” *Journal of the Franklin Institute*, Vol. 342, 2005, pp. 175-192.

14. Bose, T. and D.P. Brown, “Limit Cycles due to Roundoff in State-Space Digital Filters,” IEEE *Transactions on Acoustics, Speech, and Signal Processing*, Vol. 39, No. 4, Aug. 1990, pp. 1460-1462.

15. *CAN in Automation*. 2007, 8 Nov. 2007 <http://www.can-cia.org/>.

16. Bosch, Robert GmbH, “C_CAN Users Manual Revision 1.2,” June 6, 2000.

17. *Battery University*. 2005, 12 Nov 2007 <http://www.batteryuniversity.com/index.htm>.

18. McDonald, Tim, and John Ambrus, “Evaluation of Power MOSFET Thermal Solutions for Desktop and Mobile Processor Power,” International Rectifier, 2002.

19. Li, Alan, Brij Mohan, Steve Sapp, Izak Bencuya, and Linh Hong, “Maximum Power Enhancement Techniques for SO-8 Power MOSFETs,” Fairchild Semiconductor AN1029, 1996.

20. Alivio, Gil, John Ambrus, Tim McDonald, and Richard Dowling, “Maximizing the Effectiveness of Your SMD Assemblies,” International Rectifier AN-994, 2005.

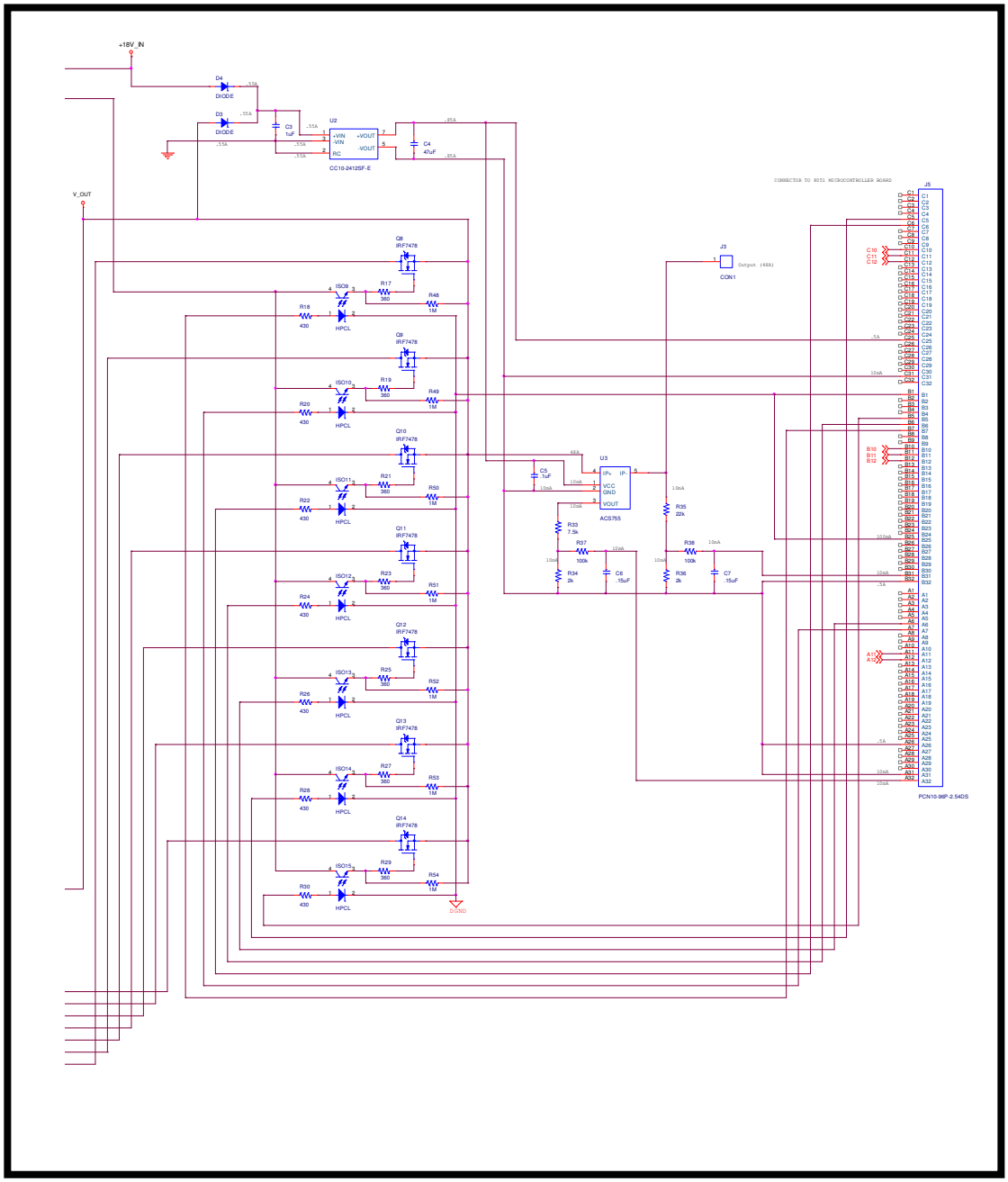
21. “C8051F040/1/2/3/4/5/6/7 Users Manual Revision 1.5,” Silicon Laboratories, Dec. 2005.

22. Gao, Lijun, and Roger A. Dougal, "Dynamic Lithium-Ion Battery Model for System Simulation," *IEEE Transactions on Components and Packaging Technologies*, Vol. 25, No. 3, Sept. 2002, pp. 495-505.

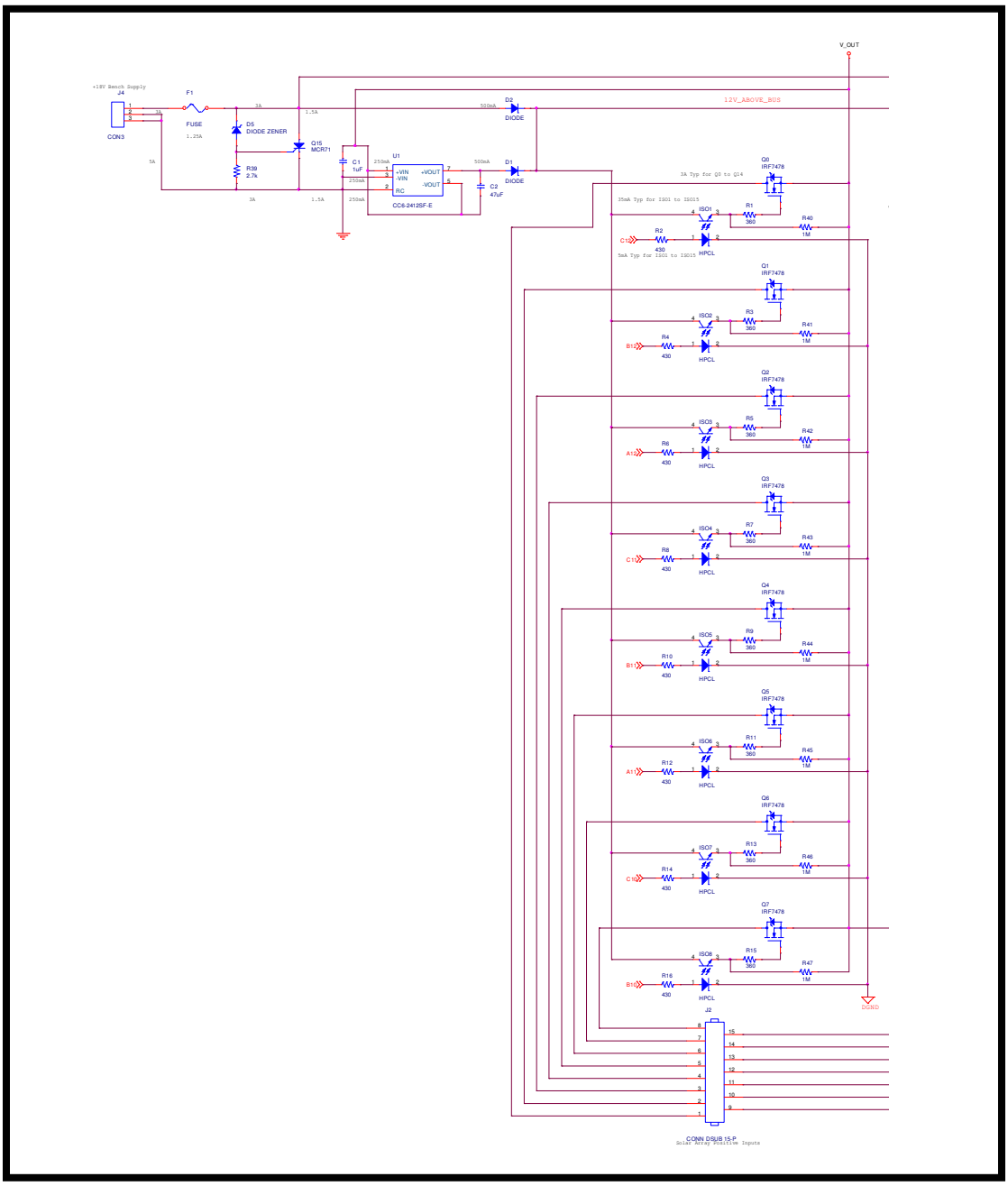
23. “Simulink Reference Discrete-Time Integrator,” Matlab Version 7.0 Help File

APPENDICES

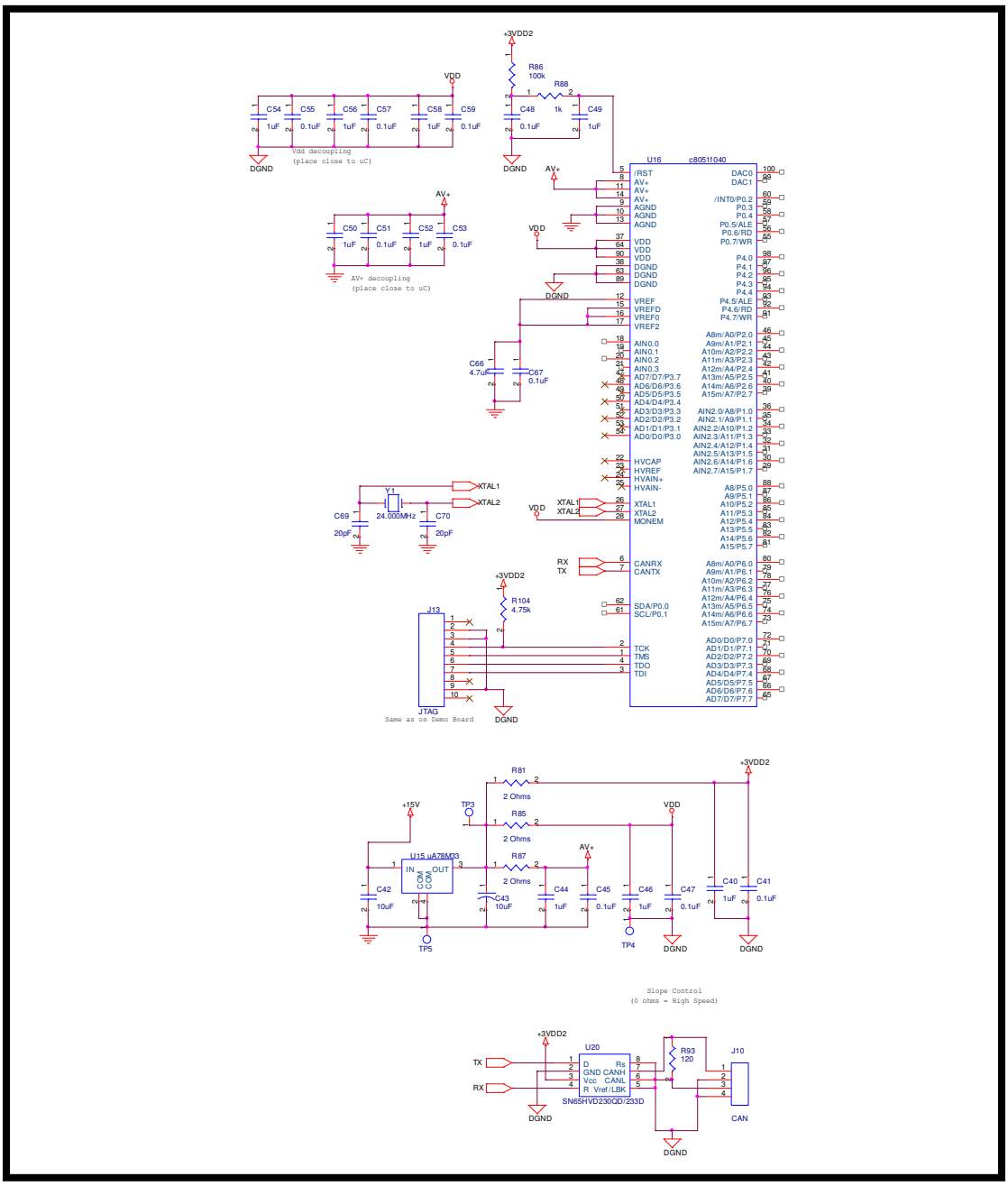
A. SASM Circuit Schematic



Sheet 1 of 3


V_OUT
12V_ABOVE_BUS
+18V Bench Supply
J4
CON3
F1
FUSE
D5
DIODE ZENER
Q15
MCR71
R39
2.7k
C1
1uF
U1
+VIN
-VIN
RC
+VOUT
-VOUT
CC6-2412SF-E
C2
47uF
D2
DIODE
D1
DIODE
Q0
IRF7478
ISO1
HPCL
R1
360
R40
1M
R2
430
C12
Q1
IRF7478
ISO2
R3
360
R41
1M
R4
430
B12
Q2
IRF7478
ISO3
R5
360
R42
1M
R6
430
A12
Q3
IRF7478
ISO4
R7
360
R43
1M
R8
430
C11
Q4
IRF7478
ISO5
R9
360
R44
1M
R10
430
B11
Q5
IRF7478
ISO6
R11
360
R45
1M
R12
430
A11
Q6
IRF7478
ISO7
R13
360
R46
1M
R14
430
C10
Q7
IRF7478
ISO8
R15
360
R47
1M
R16
430
B10
DGND
J2
CONN DSUB 15-P

Sheet 2 of 3


+3VDD2
R86
100k
R88
1k
C48
0.1uF
C49
1uF
VDD
C54
1uF
C55
0.1uF
C56
1uF
C57
0.1uF
C58
1uF
C59
0.1uF
Vdd decoupling
(place close to uC)
DGND
AV+
C50
1uF
C51
0.1uF
C52
1uF
C53
0.1uF
AV+ decoupling
(place close to uC)
U16
c8051f040
/RST
AV+
AGND
VDD
DGND
VREF
VREFD
VREF0
VREF2
AIN0.0
AIN0.1
AIN0.2
AIN0.3
AD7/D7/P3.7
AD6/D6/P3.6
AD5/D5/P3.5
AD4/D4/P3.4
AD3/D3/P3.3
AD2/D2/P3.2
AD1/D1/P3.1
AD0/D0/P3.0
HVCAP
HVREF
HVAIN+
HVAIN-
C66
4.7uF
C67
0.1uF
V1
24.000MHz
C69
20pF
C70
20pF
XTAL1
XTAL2
MONEM
CANRX
CANTX
SDA/P0.0
SCL/P0.1
TCK
TMS
TDO
TDI
RX
TX
R104
4.75k
J13
JTAG
Same as on Demo Board
DAC0
DAC1
/INT0/P0.2
P0.3
P0.4
P0.5/ALE
P0.6/RD
P0.7/WR
P4.0
P4.1
P4.2
P4.3
P4.4
P4.5/ALE
P4.6/RD
P4.7/WR
A8m/A0/P2.0
A9m/A1/P2.1
A10m/A2/P2.2
A11m/A3/P2.3
A12m/A4/P2.4
A13m/A5/P2.5
A14m/A6/P2.6
A15m/A7/P2.7
AIN2.0/A8/P1.0
AIN2.1/A9/P1.1
AIN2.2/A10/P1.2
AIN2.3/A11/P1.3
AIN2.4/A12/P1.4
AIN2.5/A13/P1.5
AIN2.6/A14/P1.6
AIN2.7/A15/P1.7
A8/P5.0
A9/P5.1
A10/P5.2
A11/P5.3
A12/P5.4
A13/P5.5
A14/P5.6
A15/P5.7
A8m/A0/P6.0
A9m/A1/P6.1
A10m/A2/P6.2
A11m/A3/P6.3
A12m/A4/P6.4
A13m/A5/P6.5
A14m/A6/P6.6
A15m/A7/P6.7
AD0/D0/P7.0
AD1/D1/P7.1
AD2/D2/P7.2
AD3/D3/P7.3
AD4/D4/P7.4
AD5/D5/P7.5
AD6/D6/P7.6
AD7/D7/P7.7
+3VDD2
R81
2 Ohms
+15V
TP3
R85
2 Ohms
VDD
U15 uA78M33
IN
COM
OUT
R87
2 Ohms
AV+
C42
10uF
C43
10uF
C44
1uF
C45
0.1uF
C46
1uF
C47
0.1uF
C40
1uF
C41
0.1uF
TP5
TP4
DGND
Slope Control
(0 ohms = High Speed)
U20
SN65HVD230QD/233D
TX
RX
R93
120
J10
CAN

Sheet 3 of 3

B. M-File to Find PI and PII Controllers Roots and Zeros

```
%=========================================================================
% m-file used to find the roots and zeros of the PI and PII controllers
%=========================================================================

a=2.89;     % Sets the value of a

%---------------
% PI Controller
%---------------
kp=.001;    % Set proportional gain
ki=1;       % Set integral gain

num=[1000*a*kp 1000*a*ki];          % Plant numerator
den=[1 (1000+1000*a*kp) 1000*a*ki]; % Plant denominator
d=roots(num);   % Calculate poles
e=roots(den);   % Calculate zeros

%----------------
% PII Controller
%----------------
Kp=.33;     % Set proportional gain
Ki=.0001;   % Set integral gain

num1=[(1000*a*kp) (1000*a*ki)];          % Plant numerator
den1=[1 1000 (1000*a*kp) (1000*a*ki)];   % Plant denominator
s=roots(num1);  % Calculate poles
t=roots(den1);  % Calculate zeros
```